

Data-Driven Image Restoration

Saeed Anwar

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

October 2018

© Copyright by Saeed Anwar 2018
All Rights Reserved

Declaration

I hereby declare that this submission is my own work (based on publications in collaboration with the co-authors where due acknowledgement is made) and that, to the best of my knowledge, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma at ANU or any other educational institution, except where due acknowledgment has been made.

I also declare that all sources used in this thesis have been fully and properly cited.

Saeed Anwar

2 October 2018

To my family

Acknowledgements

First and foremost, I am very thankful to Allah “The Most Merciful”. After that, I would like to express my sincere gratitude to my panel chair and CVRG group leader, Prof. Fatih Porikli, for his guidance, motivation, and support throughout my Ph.D. It has been an absolute privilege to work with Fatih, and my Ph.D. would not have been possible without his enthusiasm for research. I am very thankful to you, professor, for treating me with kindness throughout this arduous journey, and for being immensely helpful in many ways both academic and personal matters. Your guidance has been instrumental in reducing my learning time.

I am equally grateful to my primary supervisor, Dr. Cong Phuoc Huynh. Thank you for giving me valuable pointers and ideas, shaping my research and carefully reviewing all manuscripts that I produced during Ph.D. I am indebted to Dr. Cong for his continuous guidance and for his valuable time even after he moved to Amazon. I am very thankful to Cong for bearing with me during my Ph.D.

I would also like to thank the other researchers at Data61 (previously NICTA) and ANU for their feedback on my research during various seminars and reading groups. I am grateful to my colleagues at Data61 and ANU, past and present, who created an excellent research environment. Primarily, I would like to convey my gratitude to Dr. Khurram Aftab, Dr. Ahmed Sohaib (who introduced me to this unique opportunity), Dr. Salman H. Khan, Dr. Zeeshan Hayder, Dr. Thalaiyasingam Ajanthan, Dr. Arash Shahriari, Yusuf, Masoud, and Samitha. We had the very fruitful discussion about many aspects of life, science, and religion.

I graciously acknowledge and appreciate the financial support from the Australian National University, Data61 (previously NICTA) and the Australian Government for my Ph.D. research. Their scholarships and generous travel grants have allowed me to focus on my research without having to worry about financial support. I am indebted to my teachers at the University of Peshawar, Heriot-watt University, University of Girona and University of Bourgogne. I am thankful to Ehsan, Hashim, and Wajid for supporting me during my master degree.

I owe a great deal to my family. I would like to express my gratitude to my family and friends. I want to thank, my father, Khurshid Anwar, and my mother, Mushtaq Begum, for their upbringing, financial support and countless prayers all through these years. I am also indebted to my sister, my brothers and my wife for their unconditional love and support. To my daughter, Ummy Aiman and my son,

Muhammad Saad, who makes me smile now and then. This thesis is dedicated to you all.

Finally, and above all, I am profoundly grateful to the Lord for providing me with great mentors, supervisors, friends, colleagues and family. For keeping me steadfast during the confusion, the disappointment, and the doubt. I wish I could thank him enough for his blessings and favors, "So which of the favors of your Lord would you deny?" (Al-Quran, 55:13).

Abstract

Every day many images are taken by digital cameras, and people are demanding visually accurate and pleasing result. Noise and blur degrade images captured by modern cameras, and high-level vision tasks (such as segmentation, recognition, and tracking) require high-quality images. Therefore, image restoration specifically, image deblurring and image denoising is a critical preprocessing step.

A fundamental problem in image deblurring is to recover reliably distinct spatial frequencies that have been suppressed by the blur kernel. Existing image deblurring techniques often rely on generic image priors that only help recover part of the frequency spectrum, such as the frequencies near the high-end. To this end, we pose the following specific questions: (i) Does class-specific information offer an advantage over existing generic priors for image quality restoration? (ii) If a class-specific prior exists, how should it be encoded into a deblurring framework to recover attenuated image frequencies? Throughout this work, we devise a class-specific prior based on the band-pass filter responses and incorporate it into a deblurring strategy. Specifically, we show that the subspace of band-pass filtered images and their intensity distributions serve as useful priors for recovering image frequencies.

Next, we present a novel image denoising algorithm that uses external, category specific image database. In contrast to existing noisy image restoration algorithms, our method selects clean image “support patches” similar to the noisy patch from an external database. We employ a content adaptive distribution model for each patch where we derive the parameters of the distribution from the support patches. Our objective function composed of a Gaussian fidelity term that imposes category specific information, and a low-rank term that encourages the similarity between the noisy and the support patches in a robust manner.

Finally, we propose to learn a fully-convolutional network model that consists of a Chain of Identity Mapping Modules (CIMM) for image denoising. The CIMM structure possesses two distinctive features that are important for the noise removal task. Firstly, each residual unit employs identity mappings as the skip connections and receives pre-activated input to preserve the gradient magnitude propagated in both the forward and backward directions. Secondly, by utilizing dilated kernels for the convolution layers in the residual branch, each neuron in the last convolution layer of each module can observe the full receptive field of the first layer.

Contents

Acknowledgements	vii
Abstract	ix
1 Introduction	1
1.1 Image Processing	1
1.2 Image Restoration	4
1.2.1 Image Deblurring	5
1.2.1.1 Non-Blind Deblurring	7
1.2.1.2 Blind Deblurring	8
1.2.1.3 Limitation of Existing Deblurring Algorithms	9
1.2.1.4 Our Contribution to Deblurring Task	9
1.2.2 Image Denoising	10
1.2.2.1 Limitation of Existing Denoising Algorithms	11
1.2.2.2 Our Contributions to Denoising Task	12
1.3 Thesis Outline	13
1.4 Publications	14
1.4.1 Published papers	14
1.4.2 Under-review papers	14
2 Background and Preliminaries	15
2.1 Image Deblurring	15
2.1.1 Non-blind Deblurring	15
2.1.1.1 Iterative Methods	17
2.1.1.2 Image Priors	18
2.1.2 Blind Deblurring	19
2.1.2.1 Edge Priors	20
2.1.2.2 Probabilistic Priors	23
2.1.2.3 Patch Priors	24
2.1.2.4 Class-specific Priors	25
2.1.2.5 Learning with Neural Networks	27
2.2 Image Denoising	28
2.2.1 Image Filtering	28

2.2.1.1	Linear Filtering	28
2.2.1.2	Median Filtering	29
2.2.1.3	Denoising via Local Statistics	30
2.2.1.4	Bilateral Filtering	30
2.2.2	Methods using Local Structure Similarity	30
2.2.2.1	Non-local Means	31
2.2.2.2	Block Matching and Three Dimensional Filtering	31
2.2.2.3	Non-local Bayes	33
2.2.2.4	Weighted Nuclear Norm Minimization	33
2.2.2.5	Principal Component Analysis	33
2.2.2.6	Dual Domain & Progressive Image Denoising	34
2.2.3	External Denoising Methods	34
2.2.3.1	Targeted Image Denoising	35
2.2.3.2	Combined Image Denoising	35
2.2.4	Learning Patch Statistics	36
2.2.4.1	Denoising via Singular Value Decomposition	36
2.2.4.2	Non-local Sparse Models	36
2.2.4.3	Spatially Adaptive Iterative Singular Value Threshold- ing	37
2.2.4.4	Gaussian Mixture Model priors	37
2.2.4.5	Adaptive Image Denoising	38
2.2.5	Convolutional Neural Networks	39
2.2.5.1	Cascade of Shrinkage Fields	39
2.2.5.2	Trainable Nonlinear Reaction-Diffusion	39
2.2.5.3	DnCNN & IrCNN	40
2.2.5.4	Non-local Color Image Denoising with CNN	41
2.2.5.5	FormResNet	41
2.2.5.6	Wavelet Domain Deep Network	41
2.3	Summary	42
3	Image Deblurring with a Class-Specific Prior	43
3.1	Introduction	43
3.2	Problem Formulation	48
3.2.1	Image Prior	48
3.2.2	Objective Function	49
3.3	Deblurring Framework	50
3.3.1	Estimating \mathbf{w} given \mathbf{x} and \mathbf{k}	50
3.3.2	Latent Image Estimation	50
3.3.3	Blur Kernel Estimation	52

3.3.4	Implementation	53
3.3.5	Extension to Color Images	55
3.4	Results and Discussion	56
3.4.1	Datasets and Experimental Settings	56
3.4.2	Ablation Study	57
3.4.2.1	Effectiveness of the Prior	57
3.4.2.2	Influence of Data Fidelity Terms on Kernel Estimation	59
3.4.2.3	Influence of the Dataset Size	59
3.4.2.4	Choice of the Training Class	60
3.4.2.5	Schedule of the Prior Weight β	61
3.4.2.6	Number of Bandpass Filters	61
3.4.2.7	Reconstruction Error of the Latent Image	62
3.4.2.8	Grayscale vs. Color	63
3.4.2.9	Convergence	63
3.4.2.10	Runtime	65
3.4.2.11	Real-world Images	65
3.4.2.12	Reconstruction of a Cat Image	66
3.4.2.13	Distribution of weights of filtered training images	67
3.4.3	Comparisons with Generic Image Deblurring	68
3.4.4	Comparison with Exemplar-based Methods	77
3.5	Discussion	80
4	Category-Specific Object Image Denoising	83
4.1	Introduction	84
4.2	Denoising Problem Formulation	85
4.2.1	Support patch search	86
4.2.2	Transform domain formulation	87
4.2.3	Data Fidelity	88
4.2.4	Support Patch Group Membership	88
4.2.5	Low-rank Constraint	89
4.3	Optimization	89
4.3.1	Patch Denoising	90
4.3.1.1	Update of α_i with Fixed \mathbf{M}_i	91
4.3.1.2	Update of \mathbf{M}_i with Fixed α_i	91
4.3.2	Recovering Latent Images	91
4.3.3	Implementation Details	92
4.4	Experiments	94
4.4.1	Datasets and Parameter Settings	94
4.4.2	Influence of the External Dataset Size	95

4.4.3	Influence of the number of support patches	96
4.4.4	Relative Importance of the Priors	96
4.4.5	Runtime Comparisons	96
4.4.6	Role of the External Image Category	97
4.4.7	Sensitivity to Pose Variations	102
4.4.8	Comparisons with Internal Denoising Methods	102
4.4.9	Comparisons with External Denoising Methods	103
4.4.10	Robustness to Misalignments and Rotations	104
4.4.11	Extension to Color Images	104
4.5	Discussion	104
5	Chaining Identity Mapping Modules for Image Denoising	107
5.1	Introduction	107
5.2	Chain of Identity Mapping Modules	109
5.2.1	Network Design	110
5.2.1.1	Network Elements	110
5.2.1.2	Justification of the Design	111
5.2.1.3	Our Formulation	112
5.2.2	Learning to Denoise	112
5.3	Experiments	113
5.3.1	Benchmark Datasets and Baseline Methods	113
5.3.2	Training Details	113
5.3.3	Boosting Denoising Performance	114
5.3.4	Identity Mapping Modules	114
5.3.5	Ablation Study	114
5.3.5.1	Influence of the Patch Size	114
5.3.5.2	Number of Modules	115
5.3.5.3	Kernel Dilation and Number of Layers	115
5.3.5.4	Combination of Kernel Dilation, Identity Connection and Boosting	116
5.3.6	Comparisons	117
5.3.6.1	Classical Images	119
5.3.6.2	BSD68 Dataset	121
5.3.6.3	Color Image Denoising	121
5.3.6.4	Darmstadt Noise Dataset: Real-world Images	122
5.4	Discussion	124

6 Conclusion and Future Work	127
6.1 Conclusion	127
6.2 Future Directions	129
Bibliography	133

List of Figures

1.1	Examples of different kind of artifacts found in images.	2
1.2	A degradation model where the input image is corrupted by first passing the image into a filter and then adding noise to it.	4
1.3	An example of cat image blurred by eight different blur kernels. The blur kernels are shown in the left top corner of each image. The blur kernel values are positive and normalized so that the sum of its elements is unity.	6
1.4	Examples of grayscale and color images for noise variances of 10, 30, 50 and 75.	6
1.5	Example result of image deblurring. Given the blurry image on the left, we need to restore the original image shown on the right.	7
1.6	Visual artifacts caused by the direct inverse filter. On the left: blurred image and PSF and on the right output of inverse filtering.	8
1.7	An example of image denoising. Given the noisy image on the left, we need to restore the original image shown on the right.	10
2.1	The neural blind deblurring system of [Chakrabarti, 2016]. The blurry patch is decomposed into multiple frequency “bands”, where L is low pass, B_1, B_2 are band-pass and H stands for high-pass frequency components. Furthermore, DFT means discrete Fourier transform and IDFT stands for inverse discrete Fourier transform.	27
2.2	Example of grouping patches from noisy images. Each image shows a reference patch “R” (in red) and similar looking patches (in blue). . . .	32
2.3	BM3D framework. The process is repeated as indicated by the dashed lines.	32
2.4	A general framework for the external denoising methods.	34
2.5	The architecture of the TRND network. k_i^1 is the set of linear kernels, y_0 the degraded image, x is the groundtruth image and α_1 is strength of the term.	40
2.6	The architecture of the DnCNN and IrCNN network.	40
2.7	FormResNet Proposed network structure.	41

3.1	Recovering spatial frequencies that have been suppressed by a blur kernel using band-pass frequency components from the training data. .	45
3.2	A visual demonstration of the proposed prior. Top row (from left to right): original (ground-truth) image \mathbf{x}^* , input blurred image \mathbf{y} , the image reconstructed by the weighted combination of all the filtered training images, and the absolute difference $\ \mathbf{x} - \mathbf{x}^*\ $. Second row (from left to right): the four most important filtered training images sorted by the descending order of their weights (shown in the inset). Third row: the training images corresponding to those in the second row. Fourth row: the bandpass filters (shown in the frequency domain) involved in the filtered training images in the second row.	47
3.3	Latent images and kernels recovered by our method without (third column) and with the proposed prior (fourth column).	57
3.4	Influence of the data fidelity term in the objective function on the kernel estimate. A pair of PSNR/SSIM error metrics is shown for each kernel estimate in the sub-figures (d)–(f). (a) ground-truth image, (b) blurred image, (c) ground-truth kernel, (d) estimated kernel with the intensity term only, (e) estimated kernel with the gradient term only, (f) estimated kernel with both terms.	60
3.5	The relative reconstruction errors (averaged over 80 test images) for the INRIA person [Dalal and Triggs, 2005], the CMU-PIE [Sim et al., 2002] and the Yale-B [Georghiades et al., 2001] datasets.	62
3.6	The convergence of the iterative algorithm. The image and kernel similarity between the estimated and the ground-truth are measured in terms of the SSIM.	64
3.7	Estimated kernels for the sample image in Figure 3.6 at different scales. As visible, the kernel becomes progressively more similar to the ground-truth at finer resolutions.	64
3.8	Deblurring a real-world image (with no known ground-truth) from the dataset in [Shi et al., 2014].	65
3.9	Deblurring results for real input images from [Pan et al., 2014a], where the one in the first row contains noise and saturated pixels.	66
3.10	The reconstruction of a cat image, by taking the weighted combination of all the filtered training images from the Cat dataset. From left to right: blurred input image, important filtered training images, and reconstructed image.	67
3.11	The weights $w_{i,j}$ for the reconstruction of the latent image in Figure 3.2.	68

3.12	Results for a sample image from the Car dataset in [Krause et al., 2013]. The restored image from our method has more legible text on the license plate compared to the other methods.	71
3.13	Comparison of several methods on a sample image selected from the INRIA dataset [Dalal and Triggs, 2005]. Our method successfully recovers parts of the image with a significant resemblance to the ground-truth, including the pedestrians and the bus in the background. Our estimated kernel is also the most accurate among all the methods. . . .	71
3.14	Comparisons on a sample image from the Cat dataset [Zhang et al., 2008]. Our method recovers fine texture around the neck, mouth, and whiskers, which cannot be accurately reproduced by the others. . . .	72
3.15	Comparisons on a sample image with strong edges and a blurred background, selected from the ETHZ Shape Classes dataset [Ferrari et al., 2010]. The visual quality, <i>e.g.</i> sharpness of the text on the label, reproduced by our method is par to the best one among the other methods, <i>i.e.</i> [Sun et al., 2013].	72
3.16	Comparisons on a sample image with rich textures, selected from the ETHZ Shape Classes dataset [Ferrari et al., 2010]. On a magnified view, the image our method recovers is sharper than those generated by most of the methods, and comparable to the best, <i>i.e.</i> of [Xu et al., 2013], while exhibiting a less degree of ringing artifacts.	73
3.17	Comparisons on a face image selected from the CMU PIE dataset [Sim et al., 2002]. Although our deblurred image appears to be similar to those produced some other methods, its intensity profile (on the face) is richer than the other methods.	73
3.18	Comparisons on a sample face image selected from the Yale-B dataset [Georghiades et al., 2001]. The image we recover is more natural and contains less ringing and exaggerated contrast artifacts. Our estimated kernel is also the closest to the ground-truth.	74
3.19	Comparisons on a sample image from the FEI dataset [Thomaz and Giraldi, 2010]. Differences can be better seen in magnified view.	74
3.20	Intensity profiles (corresponding to pixel row 55) of the deblurred images produced by our method and others. The input blurred image is given in Figure 3.18b. The red trace in each subplot shows the ground-truth profile.	75
3.21	Comparison with [Zhang et al., 2011]. (a) Ground-truth image, (b) blurred image, (c) deblurring results produced by [Zhang et al., 2011], which is reported as a failure case in their paper, (d) our results.	77

3.22	Comparison to [Hacohen et al., 2013]’s on a blurred image taken from their paper.	78
3.23	Deblurring of an image containing foreground text and complex background. (a) Ground-truth (sharp) image, (b) blurred image, (c) deblurring results by [Pan et al., 2014b], (d) our results (zoom in to see the differences).	79
3.24	Results for an image provided by [Pan et al., 2014a]. First column: ground-truth image, second column: blurred images and original blur kernels (at the top left corners of the images), third column: deblurred images and estimated kernels by [Pan et al., 2014a], fourth column: our results.	80
4.1	Denoising results of two sample images from <i>face</i> and <i>cat</i> categories. As visible, by using the same category support dataset we generate higher PSNR scores-shown in red (best viewed in high-resolution). . . .	84
4.2	Searching and selecting support patches for a given noisy patch \mathbf{y}_i . Candidate images similar to the noisy image (measured by SSIM) are selected from the given database. Subsequently, in each candidate image, we search for patches that are similar to the noisy patch, <i>i.e.</i> within a Euclidean distance of τ from \mathbf{y}_i . The search is restricted to a local window in each candidate image. Finally, among the remaining patches, only the nearest neighbors to \mathbf{y}_i are retained for denoising. . . .	86
4.3	Denoising accuracy (in PSNR) at noise standard deviations $\sigma_n = 30$ and $\sigma_n = 50$. Left: Our method is robust to the changes in the dataset size, which has a low impact on the results. Right: Increasing the number of support patches slightly degrades the denoising results.	95
4.4	Denoising results produced by different methods for a face image in a profile view from the FEI face dataset [Thomaz and Giraldi, 2010] when $\sigma_n = 30$. Our method can denoise the input image even with a different pose from those in the noise-free dataset (Differences are better viewed with high-resolution display).	97
4.5	Denoising results produced by different methods for a face image selected from the Gore dataset [Peng et al., 2012] when $\sigma_n = 20$. Our method is able to denoise the face image even with a different pose from those in the noise-free dataset.	98
4.6	Visual denoising results produced for $\sigma_n = 50$, by several methods for a sample texture image from the Multiview dataset [Hirschmüller and Scharstein, 2007]. Our approach can recover much more texture details as compared to competing methods.	100

4.7	Denoising results achieved by various methods for a sample image with a noise standard deviation $\sigma_n = 30$. The ground truth image is from the Gore dataset [Peng et al., 2012].	100
4.8	Visual denoising results for a texture image selected from the Multi-view dataset [Hirschmüller and Scharstein, 2007] where $\sigma_n = 50$. Our method can recover much more texture details than the others (please zoom-in to see details).	101
4.9	Denoising results for different methods from the dataset in [Zhang et al., 2008] when $\sigma_n = 50$. The top two rows show the candidate images from the dataset that are most similar to the noisy image. . . .	101
4.10	Comparison of a few denoising methods on color images from the datasets in [Hirschmüller and Scharstein, 2007] and [Thomaz and Giraldi, 2010], where the noise standard deviations are $\sigma_n = 20$ and $\sigma_n = 80$, respectively. Our method can recover much more details than the others.	102
5.1	Denoising results for an image corrupted by the Gaussian noise with $\sigma = 50$. Our result has the best PSNR score, and unlike other methods, it does not have over-smoothing or over-contrasting artifacts. Best viewed in color on high-res display.	108
5.2	The proposed network architecture, which consists of multiple modules with similar structures. Each module is composed of a series of pre-activation-convolution layer pairs.	110
5.3	Denoising quality comparison on a sample image with strong edges and texture, selected from classical image set for noise level $\sigma_n = 50$. The visual quality, <i>i.e.</i> sharpness of the edges on the wings and small textures reproduced by our method is the better than others.	117
5.4	Comparison on a sample image from BSD68 dataset [Roth and Black, 2009] for $\sigma_n = 50$. Our network is able to recover fine textures in the background and on the castle, while other methods cannot reproduce such textures accurately.	118
5.5	Denoising performance for state-of-the-art versus the proposed method on sample color images from the dataset in [Roth and Black, 2009], where the noise standard deviation σ_n is 50. The image we recover is more natural, contains less contrast artifacts and is closest to the ground-truth.	118

5.6	A sample color image with rich textures, selected from the BSD68 dataset [Roth and Black, 2009] for $\sigma_n = 25$. On a magnified view, the image our network recovers is sharper than those generated by most of the methods	120
5.7	Real images from Darmstadt Noise Dataset (DND) benchmark for different denoising algorithms [Plötz and Roth, 2017].	123
5.8	Two real images from [Zhang et al., 2017a] denoised by our noise level agnostic color and grayscale models, respectively.	124

List of Tables

3.1	A comparison of the accuracy achieved by our deblurring framework on all the mentioned datasets with and without the proposed prior. . . .	58
3.2	Influence of intensity and gradient fidelity terms on the deblurring results.	58
3.3	Deblurring performance (in PSNR) on the CMU PIE dataset for different numbers of training images.	59
3.4	Deblurring performance (in PSNR) for different classes of the blurred input image and the external training datasets. The PSNR is significantly higher when the external dataset matches the input image category.	61
3.5	The average image accuracy (in PSNR) achieved with a constant prior weight β when our algorithm is evaluated on the Person dataset [Dalal and Triggs, 2005].	61
3.6	The average accuracy of the deblurred image (in PSNR) for the Person dataset [Dalal and Triggs, 2005], with respect to different numbers of bandpass filters M	62
3.7	A comparison of the image and kernel accuracy (in PSNR) obtained using greyscale vs. colour input images. The results are reported for the INRIA person dataset [Dalal and Triggs, 2005].	63
3.8	The accuracy of the deblurred images, measured by SSIM and PSNR. The missing results, indicated by “-”, occurs when the respective method is not capable of dealing with the low resolution of the input images. Best results are in bold.	69
3.9	The similarity between the estimated kernel to the ground-truth, measured by SSIM and PSNR. The missing results, indicated by “-”, occurs when the respective method is not capable of dealing with the low resolution of the input images. Best results are in bold.	70
4.1	Run-time comparisons (in seconds) on a test image of size 304×228 . . .	95
4.2	Denoising performance (in PSNR) when using different image category datasets. The PSNR is maximal when the external dataset category matches the noisy image category.	96

4.3	Performance comparison between our method and internal denoising techniques on several datasets, in terms of PSNR (in dB).	98
4.4	Performance comparison between our method and external denoising techniques on several datasets, in terms of PSNR (in dB).	99
4.5	Denoising performance in PNSR (dB) on color images for noise levels $\sigma_n = 30, 50, 70, 80, 100$. Best results are in bold.	99
5.1	Detailed architecture of an identity mapping module.	115
5.2	Denoising performance (in PSNR) on the BSD68 dataset [Roth and Black, 2009] for different sizes of training input patches for $\sigma_n = 25$, keeping all other parameters constant.	115
5.3	The average PSNR accuracy of the denoised images for the BSD68 dataset, with respect to different number of modules M . The higher the number of modules, the higher is the accuracy.	115
5.4	Denoising performance for different network settings to dissect the relationship between kernel dilation, number of layers and receptive field.	116
5.5	PSNR reported on the BSD68 dataset for $\sigma_n = 25$ when different features are added to the DnCNN baseline (first row).	116
5.6	Comparisons with state-of-the-art methods on BSD68 with $\sigma_n = 50$, and BSD100 with $\sigma_n = 25$. The results of [Bae et al., 2017] and [Jiao et al., 2017] are taken from their respective papers.	117
5.7	Performance comparison between image denoising algorithms on widely used classical images, in terms of PSNR (in dB). The best results are highlighted with bold red color while the blue color represents the second best denoising results.	119
5.8	Performance comparison between our method and existing algorithms on the grayscale version of the BSD68 dataset [Roth and Black, 2009]. The missing denoising results, indicated by “-”, occurs when the method is not trained to deal with the input noisy images.	120
5.9	The similarity between the denoised color images and the ground-truth color images of the BSD68 dataset for our method and existing algorithms measured by PSNR (in dB) reported for noise levels of $\sigma=15, 25$, and 50.	121
5.10	Mean PSNR and SSIM of the denoising methods evaluated on the real images dataset by [Plötz and Roth, 2017].	122

List of Algorithms

1	Deblurring with the class-specific prior.	54
2	Denoising with category-specific support patches	93

Introduction

Everything you can imagine is real.

Pablo Picasso

In the current digital age, camera-based equipments, such as smartphones and hand-held cameras allow people to capture a significant amount of image data and help share it through social media. However, the image data quality suffers from various forms of artifacts and degradations such as blur (motion, defocus *etc.*) and noise (Gaussian, speckle, thermal *etc.*). The process of undoing the artifacts and recovering image details is termed as image restoration. Image restoration is a challenging and an ill-posed problem. However, we examine whether it is possible to procure more image contents to recover missing or corrupted observed image details. Fortunately, the amount of data available in our modern digital age changes the way we approach recent problems. Exemplar-based methods and learning methods are emerging and further improving the quality of image restoration tasks. For example, class-specific denoising methods are becoming popular for removing corrupted pixels and enhancing the image quality. Similarly, exemplar-based image deblurring is introduced recently to demonstrate that large training sets provide superior performance. Before delving into the image restoration techniques in this thesis, we will briefly discuss common image artifacts and corresponding image enhancement techniques.

1.1 Image Processing

Image processing is used to preprocess images into a suitable form for tasks such as computational photography, recognition, and classification. Computer vision applications require utmost care in designing the image enhancement stages to achieve desirable results. Examples of image processing are color correction, color balancing, sharpening, warping, geometric transformations, removing unwanted objects, removing blur and noise reduction.

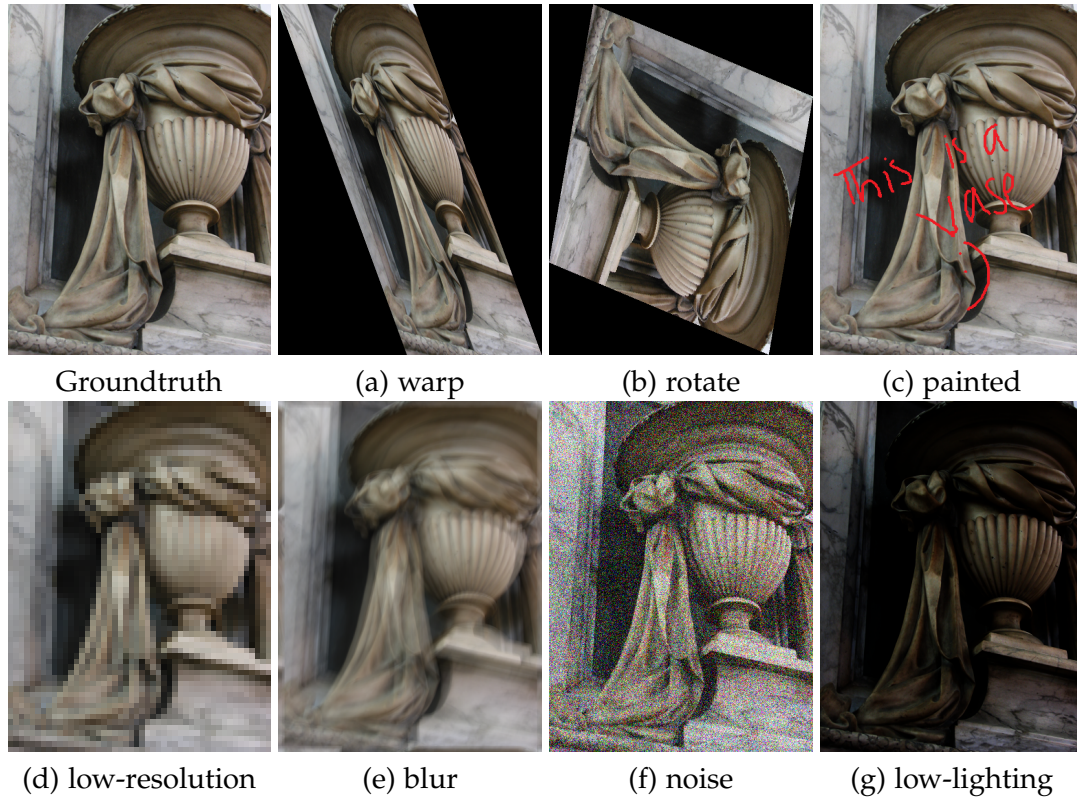


Figure 1.1: Examples of different kind of artifacts found in images.

Color correction [Rizzi et al., 2003; Vrhel and Trussell, 1992] means to modify the color of an input image so that its colorimetric properties are similar to those of a target image. Color is a vital cue topic in computer vision, human-computer interaction and feature extraction applications. The colors present in the images are due to intrinsic properties of the objects and the light source. The ability to filtered out the effects of the light source can account for color correction and is called color constancy.

Similarly, image warping [Wolberg, 1990; Glasbey and Mardia, 1998] is the process of mapping a source image into a destination image and is used for the correction of geometric distortions due to imperfect imaging sensors. Sometimes artifacts like pincushion or barrel distortion are introduced by camera lenses, while other distortions are also possible such as projective distortions introduced by perspective views. The process of removing these geometric distortions from an image is known as image warping in computer vision.

Furthermore, the process of recovering missing information or restoring the damaged areas in an image is known as image inpainting [Bertalmio et al., 2000; Criminisi

et al., 2004]. This line of research is prevalent in recent years and is helpful in many applications such as recovering images from overlays *e.g.* scratches, and text, removing unwanted objects and disocclusion in image-based rendering. Similarly, image compositing and matting techniques are for editing and enhancing visual effects of images. The process of extracting an object from an image foreground is known as matting while putting in another it into the image is called compositing.

Moreover, image super-resolution [Freeman et al., 2002; Kim et al., 2016a] refers to the process of obtaining a high-resolution image from a low-resolution image. This line of research has been very active over recent years, boosted by numerous applications: Iris recognition, medical imaging, fingerprint image processing, text image enhancement and satellite imaging *etc.*

Further, the act of recovering a clean signal from a blurry one is referred as image deconvolution or image deblurring [Fish et al., 1995; Ayers and Dainty, 1988]. It is well studied for decades and is useful in many fields of image processing such as robotic vision, surveillance, object segmentation and object recognition *etc.* We present more on image deblurring in Section 1.2.2

The removal of random fluctuations of the pixels of an image is known as image denoising [Lee, 1980; Tikhonov et al., 1977]. Image denoising is a preprocessing step in many areas of computer vision *e.g.* medical imaging, satellite imaging, ultrasound imaging, infrared imaging and astronomical images *etc.* We discuss denoising in detail in the Section 1.2.2

In image processing, segmentation [Shi and Malik, 2000; Haralick and Shapiro, 1985] means partitioning of an image into many segments. More specifically, segmentation is the process of assigning the same label to the pixels that have specific similar properties in an image. Segmentation aims to simplify and modify the representation of an image into the more meaningful way which is straightforward to investigate. Commonly, segmentation is employed in locating objects and boundaries in an image.

Another worth mentioning image processing technique is edge detection [Perona and Malik, 1990; Canny, 1987]. Edge detection aims to identify the boundaries or sets of pixels in an image where the pixels of an image change abruptly or has discontinuities. Edge detection is useful in many applications such as segmentation, data extraction *etc.*

Image processing can be classified into two principal categories: image restoration and image enhancement. Image restoration restores an image by removing artifacts such as noise, blur, and scratches *etc.* while image enhancement deals with sharpening the characteristics of an image, for example, adding color, details or contrast to an image. In other words, image restoration is related to recovering original

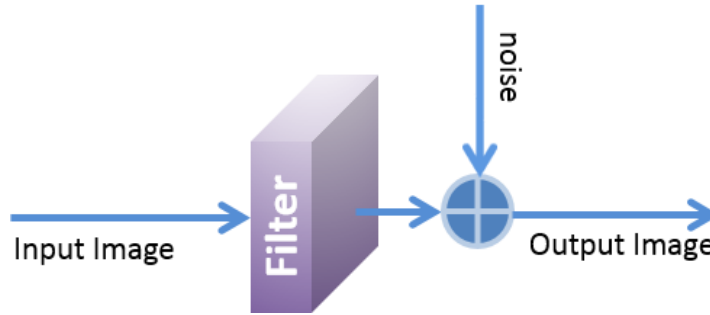


Figure 1.2: A degradation model where the input image is corrupted by first passing the image into a filter and then adding noise to it.

image from a degraded or corrupted one, while on the other hand, image enhancement seeks to improve the perceptual quality of the image. Notably, the image restoration aims to precisely recover the original image even if the outcome may not be perceptually pleasing. While the image enhancement goal is to improve the visual quality of the picture, irrelevant whether the original signal becomes unwarranted. As a result, image restoration requires typically a model that describes the image formation process that how it is corrupted. However, image enhancement does not necessarily demand a model that gives the desired output. The respective underlying goals of image restoration and image enhancement is different; yet, there do exist a significant overlap between the two methods. Figure 1.1 shows images with different kinds of artifacts. In the next section, we introduce the image restoration tasks, addressed in this thesis and show in subsequent chapters how they can be formulated using external datasets.

1.2 Image Restoration

Image restoration is an old classical problem where the aim is to recover the original clean signal from its degraded observation. Although this problem is very old and well-studied, still it is relevant in many fields *e.g.* medical imaging, surveillance, robotic vision *etc.*

To this end, we consider a simple image degradation model. Figure 1.2 is an example degradation model. The image captured using the camera is a degraded version of the original scene due to various factors such as limitations of the digital camera system or external influence *e.g.* environment and human hands movement. The degradation model shown in Figure 1.2 is linear; however, more complex degradations models can also be found.

The input image in Figure 1.2 is the original image convolved with the blur filter

and then further degraded by the noise, resulting in final blurry, noisy output image. The blurring filter is also known as point spread function (PSF) or blur kernel. The blur kernel can be either uniform *i.e.* same blur kernel is convolved with every pixel of the image, or non-uniform *i.e.* different blur kernels are applied on different pixels in the same image. Some common types of blur are motion blur and defocus blur.

In addition to blur, the noise \mathbf{n} plays a role in degrading the image quality as well by randomly changing the intensity of image pixels. Common examples of noise are salt & pepper noise, speckle noise, shot noise, Gaussian noise and quantization noise. The characteristics of each noise mentioned arise from their respective noise source. Mathematically, Figure 1.2 can be modeled as

$$\text{vec}(\mathbf{y}) = \mathbf{K}\text{vec}(\mathbf{x}) + \text{vec}(\mathbf{n}), \quad (1.1)$$

where vec is an operator to convert a matrix to its vector form, \mathbf{x} is the original uncorrupted image while $\text{vec}(\mathbf{x}) \in \mathbb{R}^m$. Similarly, \mathbf{y} is the captured degraded image while $\text{vec}(\mathbf{y}) \in \mathbb{R}^m$. \mathbf{K} is the transformation matrix and \mathbf{n} is the random noise. Image restoration aims to recover the original uncorrupted image \mathbf{x} from the degraded version \mathbf{y} . If the noise in Equation 1.1 becomes zero *i.e.* $\mathbf{n} = 0$, then the image restoration problem becomes image deblurring problem, and the observed image is termed as a blurry image. Figure 1.3 shows a cat image blurred by different eight blur kernels. On the other hand, if the transformation operator becomes identity, *i.e.* $\mathbf{K} = \mathbf{I}$, the problem reduces to image denoising one. Figure 1.4 shows gray and color images for Gaussian at different noise levels. In this thesis, we are tackling the image denoising and image deblurring problems individually as opposed to adding both the degradation to a single image.

1.2.1 Image Deblurring

An enormous amount of research has been dedicated to removing blurry effects such as motion, defocus, rotational and aberrations from the captured images. According to [Baker and Kanade, 2002], the original photograph cannot be recovered by just inverting the imaging conditions. Therefore, as pointed out by [Joshi et al., 2009] that main challenge is to incorporate prior knowledge and formulate solutions which is most likely and avoid unwanted ones. The main reasons for image blur in consumer photography are object motion, camera-shake and camera-focus. Image deblurring can be broadly categorized into two cases: blind and non-blind. See Figure 1.5 as a deblurring example.



Figure 1.3: An example of cat image blurred by eight different blur kernels. The blur kernels are shown in the left top corner of each image. The blur kernel values are positive and normalized so that the sum of its elements is unity.

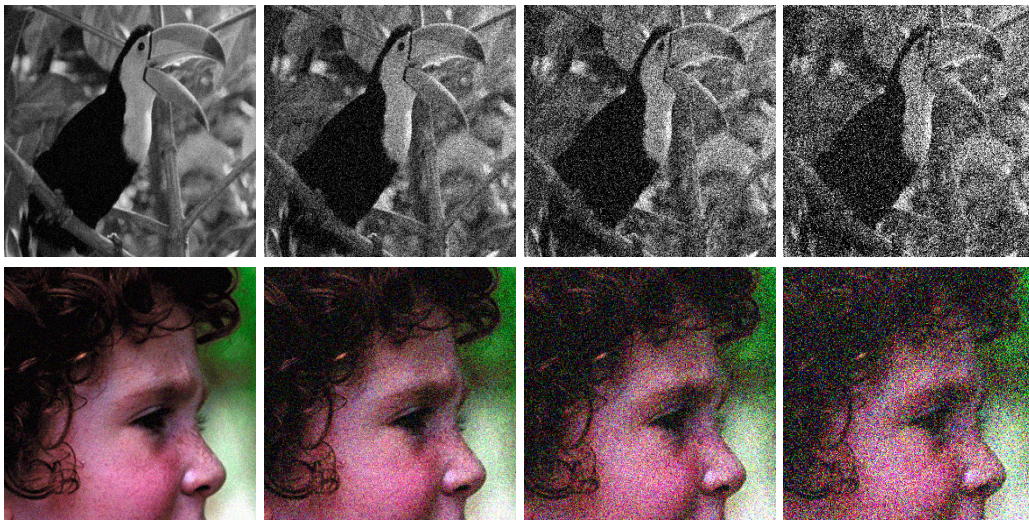


Figure 1.4: Examples of grayscale and color images for noise variances of 10, 30, 50 and 75.

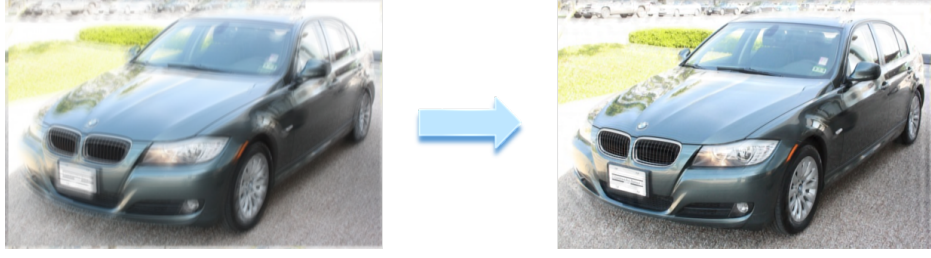


Figure 1.5: Example result of image deblurring. Given the blurry image on the left, we need to restore the original image shown on the right.

1.2.1.1 Non-Blind Deblurring

In non-blind deblurring, the aim is to recover the original unblurred image under the assumption that the blur kernel is known beforehand. In non-blind deblurring, the objective is to reduce the effect of inherent problems such as ringing artifacts, noise suppression, and increasing efficiency. Theoretically, a blurry image is modeled as a filtered version of the latent image and can be formulated as

$$\mathbf{y} = \mathbf{k} * \mathbf{x}, \quad (1.2)$$

where \mathbf{k} is the point spread function/blur kernel and “ $*$ ” is the convolutional operator. In image deblurring, for the sake of ease, computations are performed in frequency domain, as the convolution theorem states that Fourier transform F of a convolution is the element-wise multiplication, hence,

$$F(\mathbf{y}) = F(\mathbf{k}) \cdot F(\mathbf{x}), \quad (1.3)$$

in simple case, the latent image \mathbf{x} can be recovered by inverting the convolution process and can be expressed as

$$F(\mathbf{x}) = F(\mathbf{y}) / F(\mathbf{k}), \quad (1.4)$$

this process is called direct inverse filtering and will only work if there are no zero values or only small values in $F(\mathbf{k})$; otherwise, it will produce severe ringing artifacts.

An example of the deblurring using direct inverse filter *i.e.* Equation 1.4 is given in Figure 1.6. Ringing artifacts in unblurred images may be due to many reasons. Firstly, the inversion of the blur kernel may not be present. Secondly, blur kernels are band limited with zeros values at high-frequency spectrums. Thirdly, imaging system limitations such as saturation, noise accumulation, quantization error and non-linear camera function. These phenomena make image deconvolution more complex

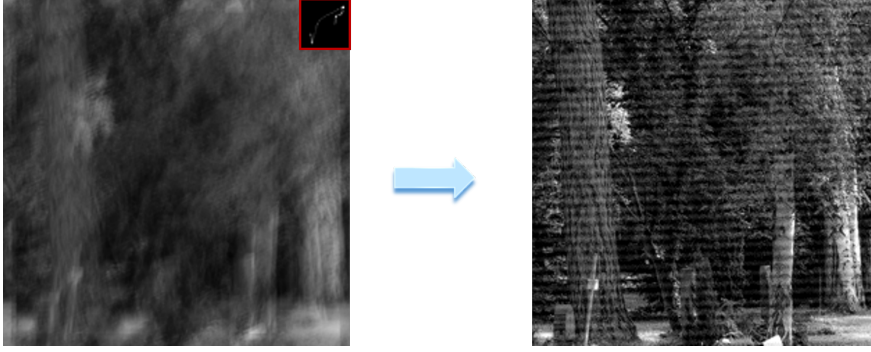


Figure 1.6: Visual artifacts caused by the direct inverse filter. On the left: blurred image and PSF and on the right output of inverse filtering.

and result in more complex forms.

Usually, image deblurring algorithms minimize two expressions: the data fidelity term and the prior term (also known as regularization). The data fidelity term corresponds to likelihood in probability and minimizes the difference between the convolved latent image and the blurry image. Commonly, the probability likelihood is equated to the distance function *i.e.* the ℓ_2 -norm

The regularization or prior term is different for different deblurring methods *e.g.* some methods apply sparsity, and others use incorporate edges in the prior. The prior term constrains the latent image \mathbf{x} . A balancing weight is used between the data fidelity and prior terms. In Chapter 2, we discuss state-of-the-art non-blind deblurring methods with their respective strengths and weakness.

1.2.1.2 Blind Deblurring

Blind deblurring techniques require prior assumptions on the blur kernel \mathbf{k} and the latent image \mathbf{x} . Blind deblurring solves the problem of motion deblurring by estimating both the blur kernel \mathbf{f} and the latent image \mathbf{x} . Numerous methods estimate the blur kernel and the latent image in an alternating optimization scheme, for example, [Pan et al., 2014a; Xu et al., 2013]. Blind deblurring is more difficult than non-blind deblurring because of more unknowns *i.e.* the blur kernel \mathbf{k} and the latent image \mathbf{x} . Defining criteria for optimization is quite difficult as different blur kernels \mathbf{k} can be estimated depending on the values of latent image \mathbf{x} and noise \mathbf{n} .

Blind deblurring shares the two entities with non-blind deblurring *i.e.* the data fidelity term and the prior term on the latent image while also introducing a new term which is the function of the blur kernel. Similarly, instead of one balancing weight, two are associated with each prior. The data fidelity function can be either based on

the gradients or intensity of the latent image and the blur kernel. In Chapter 2, we examine state-of-the-art blind deblurring approaches concerning their model design and solver construction.

Image deblurring also moved from generic methods to more specific image type deblurring. Examples of specific types are domain knowledge [Pan et al., 2014a], specific-priors [Sun et al., 2014a] and CNN [Sun et al., 2015]. Our contribution to image deblurring can be categorized as class-specific image deblurring. Detailed literature about the specific type image deblurring is provided in Chapter 2.

1.2.1.3 Limitation of Existing Deblurring Algorithms

In this section of the chapter, we point out some of the limitations of the existing deblurring methods, and in the next section, we present our contribution and how our methods avoid these limitations.

Existing image deblurring algorithms rely on the blurred image properties such as sparsity, sharp edges and heuristic filters. The sparsity priors may degenerate the solution, producing the delta blur kernel and latent image same as the blurred image. Similarly, a typical failure mode for edge-based methods is dealing with the large-scale blur present in the image induced by the large blur kernels. Also, these methods rely on image filters which are unstable and can steer the image deblurring approach to wrong solutions. However, a major problem in image deblurring is to restore distinct spatial frequencies which have been attenuated by the blurring kernel. Existing techniques usually rely on generic image priors. But, these priors only help recover part of the frequency spectrum, such as the frequencies near the high-end.

1.2.1.4 Our Contribution to Deblurring Task

Here, we present novel algorithms for image deblurring and address the limitation of the previous algorithms. Our solutions are more effective in both qualitative and quantitative terms than the competitive methods. We give our contribution and provide a brief overview that how the limitations of existing algorithms are addressed in the following paragraphs.

We introduce class specific prior for image deblurring to recover reliably distinct spatial frequencies that have been suppressed by the blur kernel. Currently, state of the art image deblurring algorithms rely on image priors *e.g.* using sparsity priors including image gradients and edges. However, such priors can only recover part of the frequency. On the other hand, we explore the potential of a class-specific image prior for recovering spatial frequencies attenuated by the blurring process.

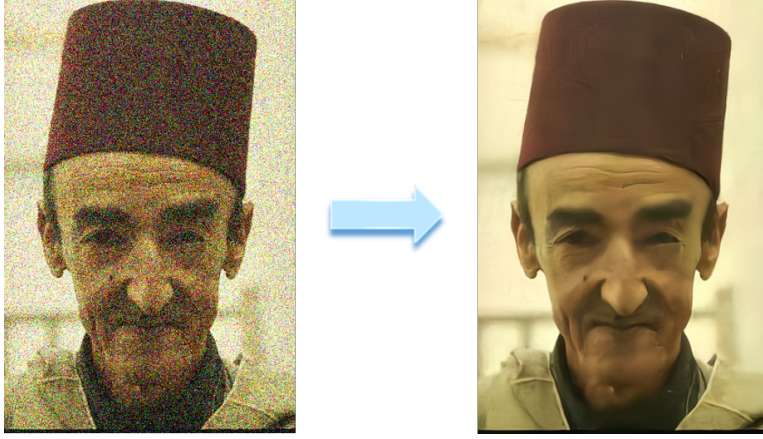


Figure 1.7: An example of image denoising. Given the noisy image on the left, we need to restore the original image shown on the right.

Specifically, we devise a prior based on the class-specific subspace of image intensity responses to band-pass filters. We learn that the aggregation of these subspaces across all frequency bands serves as a good class-specific prior for the restoration of frequencies that cannot be recovered with generic image priors.

1.2.2 Image Denoising

Image restoration becomes image denoising when \mathbf{K} becomes an identity matrix, then Equation 1.1 can be rewritten in its classical image denoising form, given an additive i.i.d. Gaussian noise model,

$$\text{vec}(\mathbf{y}) = \text{vec}(\mathbf{x}) + \text{vec}(\mathbf{n}), \quad (1.5)$$

here, aim is to recover the clean image $\text{vec}(\mathbf{x}) \in \mathbb{R}^m$ from the noisy image $\text{vec}(\mathbf{y}) \in \mathbb{R}^m$, where $\text{vec}(\mathbf{n})$ denotes the additive Gaussian noise with zero mean vector and σ^2 variance *i.e.* $\text{vec}(\mathbf{n}) \sim N(0, \sigma^2) \in \mathbb{R}^m$. Figure 1.7 is an example for image denoising.

In Chapter 4 and Chapter 5, we address the image denoising using external images. Similar to image deblurring, image denoising is also studied for decades; however, it is still relevant as it is useful in many other many computer vision tasks such as object detection, classification, and tracking. Image denoising is also useful in many other image processing tasks, for example, image deblurring, image super-resolution, and image inpainting.

The introduction of noise in images may be due to many reasons, but two leading causes stand out among all, which are electronic and shot noise. Electronic noise major causes are voltage instability, electronic components abrupt temperature

variations and analog to digital conversion. All these can be modeled as Gaussian distributed noise. Similarly, the random arrival of the photon on image sensor causes shot noise, and typically, it is modeled as a Poisson distribution. This noise is very challenging in low light condition.

Image denoising literature reveals that the noise is usually modeled as Additive White Gaussian Noise (AWGN) with zero mean due to two main reasons. Firstly, the Gaussian noise is practically applicable to other types of noises such as shot noise, as this can also be transformed into Gaussian noise using Anscombe root transformation [Anscombe, 1948]. Secondly, Gaussian distribution can facilitate the mathematical analysis as it is mathematically tractable.

With recent advancement in image denoising, researchers have started to investigate external priors [Zoran and Weiss, 2011; Yue et al., 2015] as opposed to internal priors [Dabov et al., 2007b; Buades et al., 2005]. The difference between external and internal stems from the fact that whether the reference patches are taken from the image itself or an external database. It is observed that internal priors are efficient and computationally less expensive whereas external priors achieve better performance. Our method discussed in Chapter 4 falls in external prior category.

Recently, image denoising also started taking advantage of CNN. The input to the network is a noisy observation while the target is the original clean image. Many works [Zhang et al., 2017a; Lefkimmiatis, 2016] are presented in this category and are growing to date. In Chapter 5 of this thesis, we discuss image denoising using CNN, and aim to take advantage of CNN and use external images to train our network and learn an external prior in a systematic way.

1.2.2.1 Limitation of Existing Denoising Algorithms

Here, we present common limitation of the existing state-of-the-art methods and then conclude image denoising in this chapter with our contributions to overcome the mentioned limitations.

- Many state-of-the-art algorithms [Dabov et al., 2007b; Buades et al., 2005] rely on internal self-similar patches to denoise the image. There are two main challenges to image denoising: 1) internal image denoising methods are reaching its optimal performance [Levin and Nadler, 2011; Chatterjee and Milanfar, 2012], 2) the patches that rarely occur in the image, this “rare patch” effect causes the performance to decrease.
- Recently, CNN models are employed in image denoising. Undoubtedly, CNN based image denoising methods have proved to be superior regarding performance compared to the state-of-the-art classical methods. However, these

approaches still rely on the hyper-parameter settings, extensive fine-tuning, nonlocal self-similar patches, stage-wise training and learning noise pattern without exploiting the underlying structure. These elements impede the performance of CNN based image denoising.

1.2.2.2 Our Contributions to Denoising Task

To address and overcome the limitation of previous algorithms, we introduce new novel algorithms for image denoising. Our algorithms show superior performance compared to current competitive methods. In the following paragraphs, we list our contributions and state how we addressed the shortcomings of the competitive techniques.

- We present a novel category-specific image denoising algorithm that exploits patch similarity between the input image and an external dataset only. We rely on external images in the same category as the input, to denoise textured regions. The external denoising component estimates the latent patches using the statistics, *i.e.* means and covariance matrices, of external patches, subject to a low-rank constraint. We show that our algorithm let us handle of a large variety of categories.
- We propose to learn a fully-convolutional network model for image denoising. Our denoising model learns the noise with the underlying patch structure. Also, we do not require stage-wise training and hyper-parameter setting. Our denoising network possesses distinctive features that are important for the noise removal task.
 - Each residual unit employs identity mappings as the skip connections and receives pre-activated input to preserve the gradient magnitude propagated in both directions.
 - Utilizing dilated kernels for the convolution layers in the residual branch, in other words within an identity mapping module, each neuron in the last convolution layer can observe the full receptive field of the first layer.

We have evaluated these novel algorithms on a number of datasets. We provide an extensive experimental evaluation at the end of each chapter which confirms that all our solutions surpass the performance of existing state-of-the-art methods quantitatively and qualitatively.

1.3 Thesis Outline

The structure of this thesis aims to provide the background and our contribution to image restoration as well as some future directions. The remaining chapters are summarized and organized as follows.

Chapter 2

This chapter reviews the literature on image deblurring and image denoising. Furthermore, we also provide detail of state-of-the-art algorithms and associated theories in this chapter as well as explain preliminary hypotheses that are necessary to the understanding of the remaining episodes.

Chapter 3

In this chapter, we devise a class-specific prior using the band-pass filter responses of clean, sharp images and incorporate it into a deblurring strategy. More specifically, we show that the subspace of band-pass filtered images and their intensity distributions serve as useful priors for recovering image frequencies that are difficult to recover by generic image priors. Here, we present the contribution from [Anwar et al., 2015] and its extended version [Anwar et al., 2018].

Chapter 4

In this chapter, we present image denoising algorithm that uses external, category specific image databases. In contrast to existing external image denoising that searches for patches either from a generic database or the input image, we approximate the denoised image using external non-convex priors. In this chapter, we also highlight the relationship between noisy patches and its external counterparts and the effects of external reference patches on image denoising. This chapter is based on our published works [Anwar et al., 2017c] and [Anwar et al., 2017b].

Chapter 5

In this chapter, we present a learning method for image denoising by training a CNN on a large image dataset. This chapter outlines the work presented in [Anwar et al., 2017a] and explore the power of the convolutional neural network to learn a denoising model from the natural images for denoising purposes. We show how the network learns the relationship between image noise and the image structures to provide superior image denoising results.

Chapter 6

In this final chapter of the thesis, we summarize of the works presented and discuss possible future research directions.

1.4 Publications

The contributions presented in this thesis have either been published or under review at the following venues.

1.4.1 Published papers

- **S. Anwar**, C. P. Huynh and F. Porikli, "Image Deblurring with a Class-Specific Prior," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018
- **S. Anwar**, F. Porikli and C. P. Huynh, "Category-Specific Object Image Denoising," IEEE Transactions in Image Processing (TIP), 2017.
- **S. Anwar**, C. P. Huynh and F. Porikli, "Combined Internal and External Category-Specific Image Denoising," British Machine Vision Conference (BMVC), 2017.
- **S. Anwar**, C. P. Huynh and F. Porikli, "Class-specific image deblurring," IEEE International Conference on Computer Vision (ICCV), 2015.

1.4.2 Under-review papers

- **S. Anwar**, C. P. Huynh and F. Porikli, "Chaining Identity Mapping Modules for Image Denoising," IEEE Transactions in Image Processing (TIP), 2018.

Background and Preliminaries

In this chapter, we provide a comprehensive overview of the ideas and theories of image deblurring and image denoising, both these problems are long-standing and well-researched, with algorithms too many to discuss in this literature review. We here present an up-to-date overview of state-of-the-art and the main research trend to date. Our intention in this exposition is to provide the reader the background material about the work done in this area of research. More detailed and exhaustive literature for image deblurring, though not complete can be found in [Rajagopalan and Chellappa, 2014]. Similarly, for image denoising the choice to obtain an exhaustive overview of the algorithms would be [Lebrun et al., 2012] and [Milanfar, 2013].

In this section of the thesis, we first introduce the necessary methods for image deblurring methods which can be classified into 1) blind deblurring, and 2) non-blind deblurring. Next, in the remaining chapter, we present image denoising methods which can be further categorized as 1) image filtering, 2) internal denoising, 3) external denoising and 4) learning methods.

2.1 Image Deblurring

As discussed previously, image deblurring can be divided into two main categories depending on the availability of the blur kernel *i.e.* non-blind and blind. In blind deblurring, one has to estimate both kernel and latent image while in non-blind case kernel is assumed to be available beforehand. In this section of the chapter, we first provide state-of-the-art non-blind image deblurring algorithms followed by blind ones.

2.1.1 Non-blind Deblurring

Successful and advanced state-of-the-art non-blind image deblurring dates back to late twentieth century. Examples of these methods are Wiener deblurring [Wiener,

1949], least square filtering by [Miller, 1970; Tikhonov et al., 1977], Richardson-Lucy [Richardson, 1972] and recursive Kalman [Woods and Ingle, 1981]. An exhaustive review of these early methods are provided in [Hunt, 1973].

Usually, non-blind approaches are composed of two expressions: the data fidelity term and the regularization term. The data fidelity term corresponds to likelihood in probability and minimizes the difference between the convolved latent image \mathbf{x} with the blur kernel \mathbf{k} ($\mathbf{k} * \mathbf{x}$) and the blurry image \mathbf{y} and can be expressed as

$$E_d = \Psi(\mathbf{k} * \mathbf{x} - \mathbf{y}), \quad (2.1)$$

where Ψ is a distance function and a common representation is the ℓ_2 -norm *i.e.* $\Psi(\cdot) = \|\cdot\|_2^2$ similar to [Wiener, 1949]. It is also known as Gaussian likelihood.

The regularization or prior term is different for different methods and can be presented as $\Phi(\mathbf{x})$. The regularization is essential to avoid undesirable solution, constraining the solution space. When we have the data fidelity term and prior term, then the original unblurred image can be estimated by minimizing the following objective function

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|_2^2 + \alpha \Phi(\mathbf{x}), \quad (2.2)$$

where α is the balancing weight between the data fidelity and the prior terms. In the following sections, we discuss recent representative state-of-the-art non-blind methods with their weaknesses, strengths, advantages, and disadvantages.

Two important regularizers used for non-blind deblurring are Gaussian and Tikhonov which are represented by $\Phi(\mathbf{x}) = \|\mathbf{x}\|^2$ and $\Phi(\mathbf{x}) = \|\nabla \mathbf{x}\|^2$, (∇ is the gradient operator applied on the intensity image) respectively. The Gaussian regularizer impose smoothness on the intensity values of image while Tikhonov enforces it on image gradients. After substituting the smoothness terms in Equation 2.2, we obtain

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \alpha \|\mathbf{x}\|^2, \quad (2.3)$$

and

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \alpha \|\nabla \mathbf{x}\|^2. \quad (2.4)$$

An important benefit of the regularizers as mentioned above is the simplicity of the equations and the existence of a closed form solution. Consider a sparse matrix \mathbf{K} generated from the blur kernel \mathbf{k} and an operator ν to convert a matrix to its vector

form, then Equation 2.3 can be expressed as

$$E = ||\mathbf{K}v(\mathbf{x}) - v(\mathbf{y})||^2 + \alpha ||v(\mathbf{x})||^2. \quad (2.5)$$

Taking Equation 2.5 expanding and setting its derivative equal to zero with respect to $v(\mathbf{x})$. The total energy of the system will become

$$\begin{aligned} E = & v(\mathbf{x})^T \mathbf{K}^T \mathbf{K} v(\mathbf{x}) - 2v(\mathbf{y})^T \mathbf{K} v(\mathbf{x}) \\ & + \alpha v(\mathbf{x})^T v(\mathbf{x}) + v(\mathbf{y})^T v(\mathbf{y}). \end{aligned} \quad (2.6)$$

To get the optimal solution, we take the derivative of Equation 2.6 and then set it to zero,

$$\frac{dE}{dv(\mathbf{x})} = 2\mathbf{K}^T \mathbf{K} v(\mathbf{x}) - 2\mathbf{K}^T v(\mathbf{y}) + 2\alpha v(\mathbf{x}), \quad (2.7)$$

$$\mathbf{K}^T \mathbf{K} v(\mathbf{x}) - \mathbf{K}^T v(\mathbf{y}) + v(\mathbf{x}) = 0, \quad (2.8)$$

$$v(\mathbf{x}) = \frac{\mathbf{K}^T}{\mathbf{K}^T \mathbf{K} + \alpha \mathbf{I}} v(\mathbf{y}), \quad (2.9)$$

where \mathbf{I} is the identity matrix and having same size as $\mathbf{K}^T \mathbf{K}$.

Next, we discuss the main non-blind deblurring methods, specifically, iterative methods and image prior algorithms.

2.1.1.1 Iterative Methods

Earlier, [Van Cittert, 1931] used an iterative solver for non-blind deblurring which can be represented as

$$\mathbf{x}^{m+1} = \mathbf{x}^m + \alpha(\mathbf{y} - \mathbf{x}^m * \mathbf{k}), \quad (2.10)$$

where m is the iteration index, α controls the speed of convergence and can be assigned manually or selected automatically. The effect of [Van Cittert, 1931] is same as direct inverse filtering where no prior to the image is used to find the final unblurred image.

Another important and extensively used iterative method is [Richardson, 1972]. According to [Rajagopalan and Chellappa, 2014], the method of [Richardson, 1972] is similar to Poisson Maximum Likelihood without employing any regularization on \mathbf{x} or \mathbf{k} . The performance of this method is superior as compared to direct inversion

methods as it reduces the noise in the final unblurred image. Typically, the method is run for a large number of iterations to converge and to achieve the passable outcome, however, if stopped midway, the outcome may be inferior. The algorithm of [Richardson, 1972] can be written as

$$\mathbf{x}^{m+1} = \mathbf{x}^m \left(\tilde{\mathbf{k}} * \left(\frac{\mathbf{y}}{\mathbf{x}^m * \mathbf{k}} \right) \right), \quad (2.11)$$

where $\tilde{\mathbf{k}}$ is the flipped/rotated version of \mathbf{k} . A few works built upon [Richardson, 1972] to improve the performance. One such method is [Yuan et al., 2008], which administered bilateral filtering at multiple scales, making [Richardson, 1972] more efficient and robust to noise.

2.1.1.2 Image Priors

In recent years, more advanced non-blind deblurring methods are proposed to deal with the visual artifacts induced by the inaccurate and unreliable estimation of blur kernels. Image priors are introduced to suppress ringing artifacts and noise present in the image because of the imaging system or the blur kernels. A common understanding about image priors is not to impose a higher penalty on wrong estimations to avoid deviated outcomes. In [Chan and Wong, 1998] the authors used sparse gradient prior *i.e.* $\Phi(\mathbf{x}) = \|\nabla \mathbf{x}\|_1$, popularly termed as total variation or Laplacian prior. The ∇ is the concatenation of first-order horizontal and vertical derivative of the image intensities *i.e.* $\delta_x \mathbf{x}, \delta_y \mathbf{x}$. The purpose of the ℓ_1 norm is to less penalize the deviant estimations as compared to ℓ_2 norms *i.e.* Gaussian priors.

Similarly, [Shan et al., 2008] also proposed a piecewise continuous prior for $\Phi(\mathbf{x})$ as

$$\Phi(\mathbf{x}) = \begin{cases} \lambda_1 |\nabla_i \mathbf{x}|, & \tau \geq |\nabla_i \mathbf{x}| \\ \lambda_2 (\nabla_i \mathbf{x})^2 + \lambda_3, & \text{otherwise} \end{cases}$$

where the parameter τ is the agreed upon value where the natural priors are joined together. Similarly, index i is the first order partial derivate in horizontal or vertical direction. λ s are the constant parameters. [Levin et al., 2007] suggested $\Phi(\mathbf{x}) = \|\mathbf{x}\|^p$, where $p > 1$ and termed it as hyper-Laplacian to achieve comparatively sharper details in final unblurred image. Furthermore, [Yang et al., 2009] used ℓ_1 data fidelity term for suppression of impulse noise, written as

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|_1 + \alpha \|\nabla \mathbf{x}\|_1. \quad (2.12)$$

Similar work is also presented by [Xu and Jia, 2010], to remove Gaussian and

impulse noise added to the image during formation. The choice of Gaussian and Laplacian likelihood is due to their robustness to noise and for being manageable in the formulation.

Other works to solve sparsely constrained non-blind deblurring include [Krishnan and Fergus, 2009] where half-quadratic splitting method [Geman and Yang, 1995] is employed. Half-quadratic splitting method introduces auxiliary variables to relax the problem and consist of two steps. In the first step, the image patches are treated as constants while the auxiliary variables are updated. In the second step, the auxiliary variables are kept fixed while the image patches are updated. The procedure alternatives for some iterations or when the difference between two consecutive steps is smaller than a threshold. The minimization problem is given by

$$\min_{\mathbf{x}} \sum_{i=1}^N \left(\frac{\alpha}{2} (\mathbf{k} * \mathbf{x} - \mathbf{y})_i^2 + \sum_{j=1}^J |f_j * \mathbf{x}|^p \right), \quad (2.13)$$

where i is the pixel index and f_j are the first order derivatives, namely, $f_1 = [1, -1]$ and $f_2 = [1, -1]^T$.

Next, [Zoran and Weiss, 2011] used a Gaussian Mixture Model (GMM) while [Hach Cohen et al., 2013] incorporated a correspondence between the blurred and the clear reference image. In another work, [Sun et al., 2014b] investigated context-specific priors to transfer mid and high frequency details from example scenes for non-blind deconvolution. In our deblurring method provided in chapter 3, we use [Levin et al., 2007] in our non-blind step to obtain the final output of the algorithm.

2.1.2 Blind Deblurring

Blind deblurring is more difficult than non-blind deblurring because of the high dimension of solution space and severely ill-posed problem as both blur kernel \mathbf{k} and latent image \mathbf{x} need to be estimated. Blind deblurring methods assume strong constraints on the latent image prior \mathbf{x} and the blur kernel prior \mathbf{k} . Defining a criteria for optimization is quite different as different blur kernels \mathbf{k} can be estimated depending on the values of \mathbf{x} and \mathbf{n} . Consider $\phi(\mathbf{k})$ is the prior for the blur kernel then it can be expressed as

$$\operatorname{argmin}_{\mathbf{x}, \mathbf{k}} \Psi(\mathbf{k} * \mathbf{x} - \mathbf{b}) + \alpha_1 \Phi(\mathbf{x}) + \alpha_2 \phi(\mathbf{k}), \quad (2.14)$$

where Ψ and Φ represents the entities already introduced in Equation 2.2 as $\|\mathbf{k} * \mathbf{x} - \mathbf{b}\|_2$ and $\|\nabla \mathbf{x}\|_p$, respectively. The parameters α_1 and α_2 are the weights while ϕ is sparse as most values of the blur kernel are zero or close to zero and ideally can be

represented as $\varphi(\mathbf{k}) = \|\mathbf{k}\|_1$. When all the three terms are combined we obtain

$$\operatorname{argmin}_{\mathbf{x}} \|\mathbf{k} * \mathbf{x} - \mathbf{b}\|_2^2 + \alpha_1 \|\nabla \mathbf{x}\|_p + \alpha_2 \|\mathbf{k}\|_1. \quad (2.15)$$

The parameter p can take values of 1, 2 or between 0 and 1. The deblurring methods are composed of these terms with some alteration, or more terms are added to Equation 2.14. However, this general form constraints all the required parameter for blind deblurring.

In the late twentieth century, blind deblurring methods focused on alternating algorithms where \mathbf{k} and \mathbf{x} are computed seriatim *e.g.* [Ayers and Dainty, 1988]. Similarly, [Fish et al., 1995] used the blind deblurring method of [Richardson, 1972] to maximize the probability. [Chan and Wong, 1998] proposed ℓ_1 norm *i.e.* total variation for both \mathbf{k} and \mathbf{x} and updated both the terms in an iterative fashion. All these methods lack the ability to handle complex blur kernels.

More research is done in blind deblurring as compared to non-blind deblurring. Hence, we will aim to present the most relevant algorithms here. In the rest of this section, we will first introduce edge priors, followed by the algorithms which seek to maximize marginal probability. Then, we will provide an overview of patch-based deblurring algorithms. In the second last part of this section, we will present a survey of class-specific deblurring methods and finally, conclude this section with neural network algorithms for deblurring.

2.1.2.1 Edge Priors

Single image deblurring methods utilizing edge information as a form of image sparsity rely on the implicit or explicit extraction of this information for kernel computation. [Shan et al., 2008] applied a general approach to alternate between estimation of the blur kernel and latent image until convergence. The blur kernel is obtained using.

$$\operatorname{argmin}_{\mathbf{k}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|_2^2 + \alpha_2 \|\mathbf{k}\|_1. \quad (2.16)$$

Furthermore, the latent image is estimated by an equation similar Equation 2.4, however, the difference lies in the norm on the prior of the latent image and can be written as

$$\operatorname{argmin}_{\mathbf{x}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|_2^2 + \alpha_1 \|\nabla \mathbf{x}\|_1. \quad (2.17)$$

Several approaches [Cho and Lee, 2009; Xu and Jia, 2010] enhance the detection and selection of strong edges via various techniques such as bilateral filtering, shock filtering, and gradient magnitude thresholding. The purpose is to extract salient

edges and suppress trivial ones for the latent image estimation during iterations. The process of shock filtering can be written as

$$\tilde{\mathbf{x}}^{m+1} = \mathbf{x}^m - \text{sgn}(\Delta \mathbf{x}^m) |\nabla \mathbf{x}^m|, \quad (2.18)$$

where Δ is the Laplacian operator while ∇ is the gradient operator. Shock filter when applied on latent image estimates remove small edges and enhances the salient edges *i.e.* step-like edges. Using shock and bilateral filtering, only a few salient features remain which guides the blur kernel estimate to the ground truth. Note that the thresholded $\tilde{\mathbf{x}}$ map is used as a substitute instead of the latent image in the blur kernel estimation step.

[Joshi et al., 2008] predicted the step edges underlying the blurred ones for the estimation of spatially varying sub-pixel point-spread functions (PSF). [Cho et al., 2011] also detected step edges in blurry images and used this information to compute the Radon transform of the blur kernel. Concern about these approaches is that wrong edges can be mistakenly selected based on only local information, due to the possible presence of multiple copies of the same edge induced by a large kernel width. Moreover, object classes with relatively limited texture details such as face and text do not usually benefit from methods using local edge information.

There have been a few notable examples of deconvolution methods that utilize image edge information for the estimation of the blur kernel. The fast deconvolution algorithm is based on the hyper Laplacian prior of [Krishnan and Fergus, 2009] and decomposition of the inverse kernels in the frequency domain into a series of 1D kernels of [Xu et al., 2014]. [Whyte et al., 2014] proposed a model to effectively reduce the ringing artifacts by merely discarding the saturated pixels, using only the non-saturated ones to estimate the blur kernel.

Specific to text image deblurring, [Pan et al., 2014b] proposed an effective ℓ_0 regularization method and employs half-quadratic splitting using both image gradients and image intensities. In [Pan et al., 2014b] case, the latent image prior is $\Phi(\mathbf{x}) = \lambda \|\mathbf{x}\|_0 + \|\nabla \mathbf{x}\|_0$ and the blur kernel prior is $\phi(\mathbf{k}) = \|\mathbf{k}\|^2$. Substituting these values in Equation 2.14, we separate the estimation of the latent image and the blur kernel as

$$\underset{\mathbf{x}}{\text{argmin}} [\|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \alpha_1 (\lambda \|\mathbf{x}\|_0 + \|\nabla \mathbf{x}\|_0)], \quad \text{with fixed } \mathbf{k}, \quad (2.19)$$

and

$$\underset{\mathbf{k}}{\text{argmin}} [\|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \alpha_2 \|\mathbf{k}\|^2], \quad \text{with fixed } \mathbf{x}. \quad (2.20)$$

Equation 2.19 contains ℓ_0 priors which is computationally expensive. To solve

this minimization problem, auxiliary variables u and g are introduced. Hence, Equation 2.19 can be rewritten as

$$\min_{\mathbf{x}, u, g} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \beta \|\mathbf{x} - u\|^2 + \mu \|\nabla \mathbf{x} - g\|^2 + \alpha_1 (\lambda \|u\|_0 + \|g\|_0), \quad (2.21)$$

where β and μ are the weights. When $\beta \rightarrow \infty$ and $\mu \rightarrow \infty$ then the \mathbf{x} solution to Equation 2.21 becomes that of Equation 2.19. The variables \mathbf{x} , u and g are treated as independent variables and are obtained iteratively. The values of u and g are initialized to zeros. In each iteration, the solution reduces to the following for \mathbf{x} (fix u, g)

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \beta \|\mathbf{x} - u\|^2 + \mu \|\nabla \mathbf{x} - g\|^2. \quad (2.22)$$

Once the latent image \mathbf{x} is available, then u and g can be solved separately and can be expressed as

$$\underset{u}{\operatorname{argmin}} \beta \|\mathbf{x} - u\|^2 + \alpha_1 \lambda \|u\|_0, \quad (2.23)$$

and

$$\underset{g}{\operatorname{argmin}} \mu \|\nabla \mathbf{x} - g\|^2 + \alpha_1 \|g\|_0. \quad (2.24)$$

The minimization problems in Equation 2.23, Equation 2.24 can be solved using pixelwise minimization technique. Thus, the solution becomes thresholding problem and can be formulated as

$$u = \begin{cases} \mathbf{x}, & |\mathbf{x}|^2 \geq \frac{\alpha_1 \lambda}{\beta} \\ 0, & \text{otherwise} \end{cases} \quad (2.25)$$

$$g = \begin{cases} |\nabla \mathbf{x}|, & |\nabla \mathbf{x}|^2 \geq \frac{\alpha_1}{\mu} \\ 0, & \text{otherwise} \end{cases} \quad (2.26)$$

This method works well with smooth surfaces but is less useful for non-uniform and highly textured areas/background. Our approach is distinguishable from all the above, as the latter only utilize generic edge priors into account, without considering class-specific spatial priors. Furthermore, these methods do not rely on external training images in addition to the input image.

Regarding constraints on the blur kernel, researchers have relied on the use of a norm to enforce the sparsity of the blur kernel. In this respect, [Krishnan et al., 2011] proposed the ℓ_1 -norm as a kernel prior similar to [Shan et al., 2008] while for

the latent image a normalized prior on image gradients and the outcome of blind deblurring can be achieved by alternatively solving

$$\operatorname{argmin}_{\mathbf{x}} \|\mathbf{k} * \nabla \mathbf{x} - \nabla \mathbf{y}\|_2^2 + \alpha_1 \frac{\|\nabla \mathbf{x}\|_1}{\|\nabla \mathbf{x}\|_2}, \quad (2.27)$$

and

$$\operatorname{argmin}_{\mathbf{x}} \|\mathbf{k} * \nabla \mathbf{x} - \nabla \mathbf{y}\|_2^2 + \alpha_2 \|\mathbf{f}\|_1. \quad (2.28)$$

Equation 2.27 normalizes the existing ℓ_1 norm by $\frac{1}{\|\nabla \mathbf{x}\|_2}$. The aim of this formulation is to achieve a smaller $\frac{\|\nabla \mathbf{x}\|_1}{\|\nabla \mathbf{x}\|_2}$ value than $\frac{\|\nabla \mathbf{y}\|_1}{\|\nabla \mathbf{y}\|_2}$ to avoid delta kernel (having one in the middle and zero else where) and the blurred image trivial solution.

2.1.2.2 Probabilistic Priors

Another approach is to adopt a probabilistic viewpoint by modeling the posterior probability of the latent image and the kernel. Ideally, the blur kernel can be estimated using conditional probability, given below

$$\mathbf{P}(\mathbf{k}|\mathbf{y}) = \int \mathbf{P}(\mathbf{x}, \mathbf{k}|\mathbf{y}) d\mathbf{x}, \quad (2.29)$$

where $\mathbf{P}(\mathbf{x}, \mathbf{k}|\mathbf{y})$ is the posterior distribution and can be expressed as

$$\mathbf{P}(\mathbf{x}, \mathbf{k}|\mathbf{y}) \propto e^{\Psi(\mathbf{k} * \mathbf{x} - \mathbf{y})} \cdot e^{\alpha_1 \Phi(\mathbf{x})} \cdot e^{\alpha_2 \phi(\mathbf{k})}. \quad (2.30)$$

The above equation is also the posterior probability of the objective function discussed in Equation 2.14 and Equation 2.15. Moreover, the problem with Equation 2.29 is the integration in the continuous form being computationally intractable over the latent image \mathbf{x} . Furthermore, if the image is discretized, still it is challenging to do marginalization as it requires to sum all the possible image values and is too expensive computationally. With this view, [Fergus et al., 2006] modeled the distribution of the latent image gradients as a mixture of zero-mean Gaussians and the distribution of the kernel elements as a mixture of exponential distributions and can be written as

$$\begin{aligned} \mathbf{P}(\mathbf{x}, \mathbf{k}|\mathbf{y}) &\approx \mathbf{Q}(\mathbf{k}, \mathbf{x}) = \mathbf{Q}(\mathbf{k})\mathbf{Q}(\mathbf{x}) \\ &= \prod_i \mathbf{Q}(\mathbf{k}_i) \prod_j \mathbf{Q}(\mathbf{x}_j). \end{aligned} \quad (2.31)$$

The first line of the above equation considers the independence between the blur

kernel and the latent image while the second line of the equation assumes there is no dependence between the pixels alleviating parameter computation.

On the other hand, [Shan et al., 2008] opted for a Maximum a Posteriori (MAP) formulation under the assumption of a Gaussian noise model. This formulation eventually leads to an objective function with norm constraints on the latent image to model the gradient sparsity and the smooth local prior of the image, and ℓ_1 -norm regularizer on the blur kernels as in Equation 2.17 and Equation 2.32. Improving upon this approach, [Levin et al., 2011b] aimed at maximizing the posterior distribution with the best kernel while marginalizing over all possible latent images. To reduce computational complexity, they tackle an approximate MAP problem with an EM-like iteration strategy. The M-step of [Levin et al., 2011b] can be efficiently solved in frequency domain as

$$\mathbb{E}_q(-\ln \mathbf{P}(\mathbf{x}, \mathbf{k}|\mathbf{y})) = \mathbb{E}_q(\|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2). \quad (2.32)$$

The E-step approximate the conditional probability and is similar to minimization of [Fergus et al., 2006] when the Gaussian regularization is employed on the latent image then it has a closed-form solution. Contrary to [Fergus et al., 2006], [Levin et al., 2011b] focused on only one blur kernel computation in the M-step which is more efficient as compared to maximum marginal probability.

2.1.2.3 Patch Priors

As an alternative, several methods [Zoran and Weiss, 2011; Sun et al., 2013; Michaeli and Irani, 2014] have employed selective information from image patches and their priors, rather than the whole image. [Zoran and Weiss, 2011] proposed patch-based image prior using GMM model, which is overly expressive *i.e.* models a wide range of phenomena including motion blur and defocus blur, and will eventually accommodate blur, causing imprecise convergence of the solution pair. [Zoran and Weiss, 2011] takes MAP approach to deblurring and is given by

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|_2^2 + \sum_i \log p(\mathbf{P}_i \mathbf{x}), \quad (2.33)$$

where \mathbf{P}_i is the patch centered around the i -th pixel. [Zoran and Weiss, 2011] can use any probabilistic prior. However, the best results are obtained using a GMM as

$$\log p(\mathbf{x}) = \log \left(\sum_{k=1}^K \pi_k N(\mathbf{x} | \mu_k, \Sigma_k) \right), \quad (2.34)$$

where μ_k is the mean and Σ_k is the covariance matrix while π_k is the mixing weights for each mixture component. A total of 2×10^6 natural image patches using $K = 200$ Gaussians $N(\cdot)$ and patch sizes of 8×8 pixels are used.

Building on the idea of [Zoran and Weiss, 2011], [Sun et al., 2013] modeled the patch-based image prior using atomic elements, namely, edges, corners, T-junctions, etc. learned from natural image datasets and artificial structures. The patch-based image prior utilized by [Sun et al., 2013] is computationally expensive, mainly due to a large number of patches used for modeling the primitive elements. [Michaeli and Irani, 2014] exploited the multi-scale patch recurrence property as a natural image prior to recover the blur kernel. The objective function used by [Michaeli and Irani, 2014] is not convex, and therefore their solution does not guarantee global optimality.

2.1.2.4 Class-specific Priors

More recently, class-specific information has been employed up to some extent for the problem of recognizing faces degraded by blur. [Nishiyama et al., 2011]’s approach aimed to determine the point-spread function in a blurred face image via a learning approach. Their method constructs frequency magnitude-based feature space from blurred images and learns the subspace spanned by all the face images blurred by the same kernel. To determine the kernel or PSF, it computes a distance measure between the feature vector of the blurred input image to the basis of each subspace and selects the subspace yielding the shortest distance. This method is restrictive since the set of blur kernels is required to be known beforehand for subspace learning. Besides, it has only been validated with simple Gaussian kernels. Instead of learning kernel-specific subspaces, our method aims to learn features specific to image classes.

Lately, [Zhang et al., 2011] proposed a joint image restoration and recognition method using a sparse representation of the training images. [Zhang et al., 2011] also employed iterative solving of the blur kernel and the latent image. The blur kernel is obtained in each iteration using Equation 2.20 while the latent image is obtained by

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{x} - \mathbf{D}\tilde{\mathbf{a}}\|^2 + \tau \sum_{l=1}^L |e_l * \mathbf{x}|^p, \quad (2.35)$$

where \mathbf{D} is the learned dictionary from training images, $\tilde{\mathbf{a}}$ is the sparse approximation of the latent image \mathbf{x} and e_l are the first order derivative filters *i.e.* $e_1 = [1, -1]$ and $e_2 = [-1, 1]^t$ while p is set to 0.5. Furthermore, the second term of Equation 2.36 enforces that the latent image can be well represented by the training dataset from which the dictionary \mathbf{D} is learned. Similarly, the last term of Equation 2.36 employs the sparsity for the latent image and can be termed as sparse regularization for a

natural image. Since the sparse representation prior implies that the training and test (blurred) images are well-aligned and cropped to the same resolution, this introduces some practical limitations. Also, the effectiveness of the method in recognizing faces from a blurred image depends on the presence of images of the same faces in the training set. Otherwise, the blurred image cannot be expressed as a sparse representation of the training images in the gallery, which is a crucial assumption of the method. Bearing in mind that such assumptions are quite restrictive in practice, we have designed our method without the above requirements.

Recently, [Joshi et al., 2010] proposed a method for personal photo enhancement, including deblurring, given examples in a photo collection. This approach requires manual annotation of face regions for the matting and segmentation of faces from input images. [Hacohen et al., 2013] tackled this problem, requiring a dense correspondence between a sharp reference image and its corresponding blurred image. To overcome the smoothness, this method introduced a new term in [Levin et al., 2007] sparse prior and termed it reconstruction prior. The new expression after introducing this prior is

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{k} * \mathbf{x} - \mathbf{y}\|^2 + \alpha_1 (\mathbf{I} - \mathbf{D}_m) \|\nabla \mathbf{x}\|^p + \alpha_2 \mathbf{D}_m \|\nabla \mathbf{x} - \nabla \mathbf{C}^{-1} \mathbf{r}_M\|^2, \quad (2.36)$$

where \mathbf{D}_m is the pixelwise weight to augment the reconstruction prior and the sparse prior. Further, \mathbf{r}_M is the transformed reference image obtained after dense mapping M between the blurry image \mathbf{y} and the reference image \mathbf{r} . Similarly, \mathbf{C} is the parametric color transformation between \mathbf{x} and \mathbf{r}_M and \mathbf{I} is the identity matrix.

[Hacohen et al., 2013]’s method produces decent results for complex kernels but has limited applications due to the strict requirements of the similar content between the reference and the blurred image. For example, it is hard to establish dense correspondences reliably between the reference image \mathbf{r} and the blurry image \mathbf{y} in the presence of motion blur, noise, and specifically occlusions.

Lately, [Pan et al., 2014a] introduced a face image deblurring method by selecting the best exemplar from a training set with the closest structural similarity to the blurred image. The salient edges from the exemplar image are denoted as $\nabla \mathbf{S}$, and are incorporated in to Equation 2.20 to yield

$$\underset{\mathbf{k}}{\operatorname{argmin}} \|\mathbf{k} * \nabla \mathbf{s} - \nabla \mathbf{y}\|_2^2 + \alpha_2 \|\mathbf{k}\|_2^2. \quad (2.37)$$

It requires manual annotations of salient features such as the eyes, mouth and lower contour of the face for each training image. Information at these locations then serves as guidance for deblurring face images. In the next section, we present the recent trend on neural network learning for image deblurring.

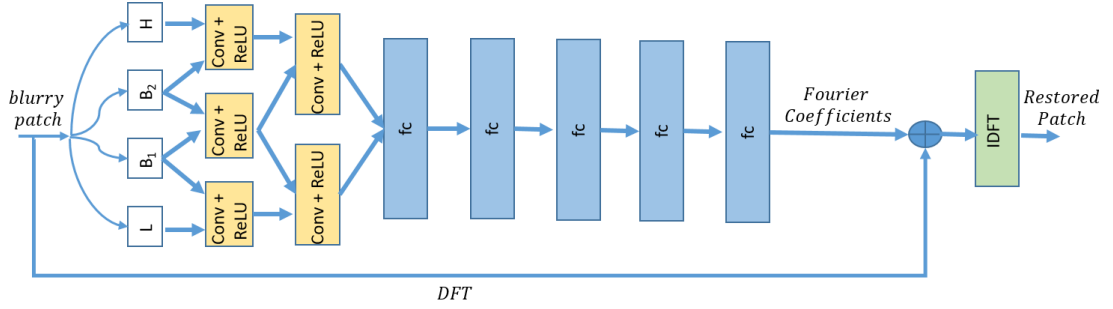


Figure 2.1: The neural blind deblurring system of [Chakrabarti, 2016]. The blurry patch is decomposed into multiple frequency “bands”, where L is low pass, B_1, B_2 are band-pass and H stands for high-pass frequency components. Furthermore, DFT means discrete Fourier transform and IDFT stands for inverse discrete Fourier transform.

2.1.2.5 Learning with Neural Networks

Besides, several works have started to pursue the “learning to deblur” approach with a significant amount of training data [Schuler et al., 2016; Chakrabarti, 2016]. Recently, [Schuler et al., 2016] proposed to learn a stack of neural networks consisting of several modules to estimate the blur kernel. This network mimics the steps of conventional iterative deblurring. Although they achieve relative success in training the system and some remarkable deblurring results; however, it failed to outperform state of the art [Sun et al., 2013]. Furthermore, the trained CNN is limited to specific kernel sizes and does not perform well when the kernel size exceeds 17×17 [Chakrabarti, 2016].

In a related development, [Chakrabarti, 2016] proposed to learn a neural network which predicts the complex Fourier coefficients to restore blurry patches of a given blurred image. The system of [Chakrabarti, 2016] used to deblur images is shown in 2.1. This idea of predicting Fourier coefficients is derived from the non-blind deblurring formulation. The deblurred patches are averaged to restore an initial estimation of the whole sharp image. Then kernel is computed from the initial estimated image as

$$k = \underset{\mathbf{k}}{\operatorname{argmin}} \sum_i ||(\mathbf{k} * (f_i * \mathbf{x}) - (f_i * \mathbf{y}))||_2^2 + \alpha ||\mathbf{k}||_1, \quad (2.38)$$

where f_i are the first and second order derivative filters. Furthermore, to obtain the final deblur image, authors use [Zoran and Weiss, 2011] as a final non-blind deblurring step. For training the neural network part, the authors generated synthetic kernels and utilized 52×10^4 patches. This method achieved remarkable results on

smooth images; however, it lags behind on images having texture.

In the next section, we present the details of the relevant literature for image denoising.

2.2 Image Denoising

Image denoising is a prevalent, well known, yet ill-posed problem in low-level vision, where the aim is to recover the clean image from its noisy version. Since the problem is under-constrained due to missing information, regularization assumptions on the noise model are taken into account such as additive white Gaussian and stationary noise. In addition, the noise values between the pixels are not correlated. Furthermore, the variance of the noise is usually assumed to be known.

During the last decade, many patch based algorithms [Buades et al., 2005; Dabov et al., 2007b, 2009; Elad and Aharon, 2006; Knaus and Zwicker, 2014, 2013; Deledalle et al., 2011; Zhang et al., 2010; Yu and Sapiro, 2011; Foi et al., 2007; Lebrun et al., 2013; Portilla et al., 2003] have been developed to improve the performance of noise removal. Nevertheless, their performance is often a marginal improvement to the BM3D method [Dabov et al., 2007b], which is still considered a widely accepted baseline even after a decade. According to [Chatterjee and Milanfar, 2010] and [Levin and Nadler, 2011], BM3D achieves near-optimal performance, close to theoretical limits on natural images. However, there is still a possibility of performance improvement of denoising using external images [Levin and Nadler, 2011; Levin et al., 2012; Chatterjee and Milanfar, 2010]. Here, we present an overview of the state of the art denoising algorithms in rest of the chapter which is divided into four broad categories, each of which is discussed in detail with the state of the art denoising algorithms in each group.

2.2.1 Image Filtering

2.2.1.1 Linear Filtering

A straightforward image denoising method is to convolve the noisy image \mathbf{y} with a Gaussian kernel \mathbf{k}

$$\mathbf{x} = \mathbf{y} * \mathbf{k}. \quad (2.39)$$

The above operation can be performed in the frequency domain as it is a linear operation.

$$F(\mathbf{x}) = F(\mathbf{y}) \cdot F(\mathbf{k}), \quad (2.40)$$

where “.” represents an element-wise multiplication *i.e.* Hadamard product. The Gaussian kernel in the frequency domain is also a Gaussian kernel. The purpose of Gaussian filtering is to attenuate high frequency (as noise is represented by high frequencies) since the Gaussian filtering is a form of low-pass filtering. Due to this reason, the images filtered with a Gaussian kernel are smoother as the high frequencies are removed.

Prior to discussing the details of the Gaussian kernel parameters, one has to know the essential property of the 2D Gaussian kernel which is separability. The one-dimensional kernel can be applied in horizontal (or vertical) direction, and then in vertical (or horizontal) direction. The resulting image has the same effect as using a 2D Gaussian kernel [Burger, 2013].

The kernel width of the Gaussian plays a vital role in noise reduction. Therefore, the width of the Gaussian kernel is determined before filtering. The value of the width depends on the intensity of the noise in the image. If the noise is high, a more substantial value for width is used and vice-versa.

Although Gaussian filtering can remove the high frequencies, however, by doing so, it has also effect on fine and sharp image structures such as edges, corners and lines *etc.* A remedy for this problem is to adapt the filter to the content of the image in hand and apply different filters to different parts of the image. This idea is further explored by bilateral filtering discussed in section 2.2.1.4

2.2.1.2 Median Filtering

Another simple approach and an alternative to linear filtering is median filtering. The median filter is non-linear and non-separable. The median filtering process simply replaces each pixel in the image with the median value in the neighbourhood centered at that pixel.

A prevailing opinion about median filtering is that it is better in preserving image features such as edges, compared to linear filtering [Caselles et al., 2000]. Simple median filtering is not better at preserving image features than linear filtering [Arias-Castro et al., 2009], however, it is better in removing outliers, therefore, often applied in case of salt & and pepper noise.

The filtering window shape and size depends on the noisy image. Popular choice for the filtering windows is square while the size of the window depends on the strength of the noise present in the image. When the noise level is high then usually several passes are applied. The first pass window is bigger than the subsequent ones. This technique of filtering in stages help in preserving the edges in the noisy image [Arias-Castro et al., 2009].

2.2.1.3 Denoising via Local Statistics

Image denoising using local statistics became popular in the late twentieth century. [Lee, 1980] proposed an image denoising algorithm using local statistics where parallel architecture can be used. This algorithm can handle both additive and multiplicative noise. However, our discussion here is restricted to the additive case only. The two assumptions are made about the distribution of noise: finite variance and zero mean.

The algorithm here assumes that the neighborhood mean, is equal to a priori mean and alike assumption is made about the variance σ^2 . Although these assumptions are debatable as pointed out by the authors in the same paper [Lee, 1980]. The following expression is used to obtain the final denoised image

$$\hat{x} = \bar{x} + \frac{(E\{(\mathbf{y} - \mathbf{y}^2)\} - \sigma^2)(\mathbf{y} - \bar{x})}{E\{(\mathbf{y} - \mathbf{y})^2\}}. \quad (2.41)$$

The above equation requires computation of two main terms: \bar{x} and $E\{(\mathbf{y} - \mathbf{y})^2\}$. \bar{x} can be easily estimated using a filter of size $m \times n$ with all ones in it. Similarly, $E\{(\mathbf{y} - \mathbf{y})^2\}$ can be obtained by the same process as \bar{x} but the filter will be applied to the pixel-wise square of the noisy image. This denoising algorithm is quite simple but useful in practice.

2.2.1.4 Bilateral Filtering

According to [Tomasi and Manduchi, 1998; Arias-Castro et al., 2009], bilateral filtering preserve edges while removing noise from the image. Bilateral filtering combines classical filtering and range-filtering. Hence, the pixels set are selected from the neighborhood and also on their related features.

In classical-filtering, the pixels are chosen based on their geometric proximity. The idea behind that the pixel values are correlated and change slowly over local neighborhood whereas noise does not follow this assumption. Therefore, the noise is attenuated in such areas, and the signal is recovered. However, this premise fails at image features such as edges. Similarly, range-filtering is performed based on photometric similarity. Range-filtering changes the gray map and suppresses unimodal histogram, hence, combining classical-filtering, and range-filtering helps preserve edges, as opposed to classical filtering which smooths the signal.

2.2.2 Methods using Local Structure Similarity

Internal image denoising with a single image is popular and usually has a low computational load. Earlier techniques focused on recovering noisy pixels from their neighboring noisy pixels e.g. Gaussian filtering, bilateral filtering, and total

variation. Later algorithms focused on re-occurrence of patches [Glasner et al., 2009] in the noisy image to reconstruct the noise-free image, examples are non-local means [Buades et al., 2005], BM3D [Dabov et al., 2007b], WNNM [Gu et al., 2014], SAIST [Dong et al., 2013], SAPCA [Dabov et al., 2009], and TSID [Zhang et al., 2010]. These algorithms are effective for areas with repetitive texture, however, on the downside, they suffer when they attempt to find corresponding matches for infrequent patches *i.e.* the patches that are rarely present in the image. To overcome this issue, some methods [Lou et al., 2009; Yan et al., 2012] proposed alternatives albeit with limited applicability. Moreover, when the noise is strong, internal denoising performance degrades drastically as it struggles to find correct reference patches.

State-of-the-art techniques in internal image denoising [Buades et al., 2005; Dabov et al., 2007b, 2009; Lebrun et al., 2013; Dong et al., 2013; Foi et al., 2007; Zhang et al., 2010] exploit repetitive local patterns that frequently occur in natural images, by selecting and grouping similar patches for denoising.

2.2.2.1 Non-local Means

The non-local means (NLM) algorithm takes benefit of the self-similarity property or redundancy in natural images. The self-similarity property states that for each patch in an image there are similar patches. This assumption is valid where the patches are in the neighborhood or within few pixels from the center of the reference patch. Hence, the redundancy in this scenario is termed as local means. Making use of this property, non-local means [Buades et al., 2005; Goossens et al., 2008] denoise images by computing a weighted average of non-local similar patches, with the weights set as the Euclidean distances between their pixel values. [Efros and Leung, 1999] have practiced the self-similarity property for texture image synthesis.

2.2.2.2 Block Matching and Three Dimensional Filtering

Block matching and three dimensional (3D) filtering is abbreviated as BM3D. Collaborative filtering with block matching and its variants [Dabov et al., 2007b; Foi et al., 2007; Dabov et al., 2009] are prominent baseline methods that exploit patch similarity in a 2D transform domain.

The core idea lies in the imposition of structural similarity among patches in each group by analyzing the subspace of the transform coefficients. The denoising of individual patches relies on the implicit assumption that insignificant coefficients correspond to the noise component and thus can be truncated via thresholding or attenuated via Wiener filtering.

BM3D considers the noisy image to have similar patches and does not rely on

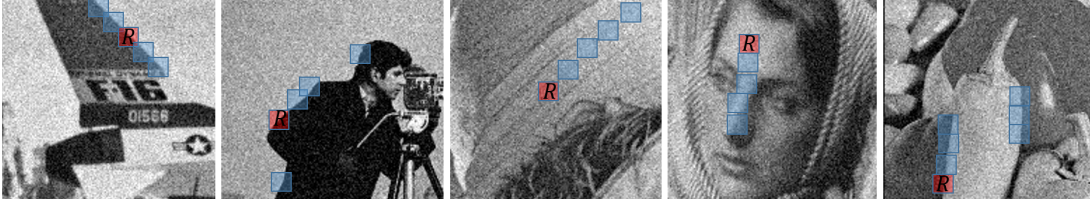


Figure 2.2: Example of grouping patches from noisy images. Each image shows a reference patch “R” (in red) and similar looking patches (in blue).

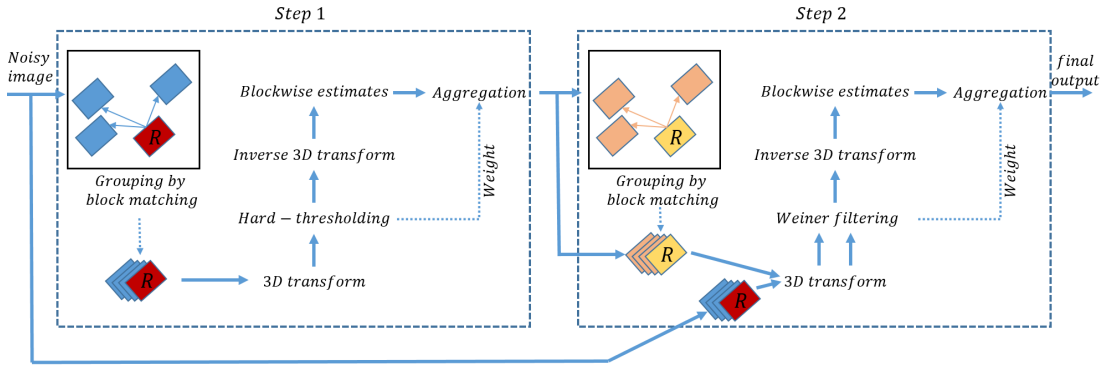


Figure 2.3: BM3D framework. The process is repeated as indicated by the dashed lines.

any regularization. BM3D searches for patches similar to a reference patch in a local neighborhood as shown in Figure 2.2 and groups them into a three dimensional (3D) block, hence, the term block-matching. The three-dimensional block is denoised, and the resulting patch is inserted into its original location. The process is repeated for every patch in the image. This process of grouping the patches, is different than clustering, as each patch can be part of multiple blocks, as opposed to clustering where each patch can only belong to one cluster.

The BM3D algorithm is a two-stage process as shown in Figure 2.3. The distinction separating the stages is how search is performed for similar patches and which procedure is employed for shrinking coefficients. In the first stage, the similar patches are searched in the noisy image, and then hard thresholding is applied to the 3D block *i.e.* the values below a certain threshold are set to zero. The outcome of this stage is a denoised image as each patch is re-inserted into their original location. In the second stage, the similar patches are searched in the resulting denoised image from the first stage. The reason for seeking patches again from the first denoised image is because it is more reliable to search similar patches using the denoised image of first step than in the noisy image itself. However, two blocks are formed,

one block consist of noisy patches while other containing patches from the denoised image of the first step. The shrinkage procedure for the second stage is Wiener filtering. The weights of noisy patches are used to approximate the block of denoised patches. In other words, the purpose of the first step is to achieve a denoised image to provide reliable similar-patches while the second phase is to get the final denoised image.

2.2.2.3 Non-local Bayes

Similar to BM3D, Non-Local Bayes (NLB) [Lebrun et al., 2013] consists of a two-step process that estimates the values of a latent patch from the mean and covariance matrix of its similar patches. To obtain an invertible covariance matrix NLB algorithm keeps a fixed number of similar patches which are more than the size of the noisy patch. NLB is also a two-step process like BM3D but having both steps identical with only using the estimate from the first step. NLB results are better in case of color images than its counterparts.

2.2.2.4 Weighted Nuclear Norm Minimization

Nuclear Norm Minimization (NNM) is becoming popular in recent years due to its ability to efficiently recover low-rank matrices. NNM uses frobenius-norm to observe the difference between the noisy image \mathbf{y} and the latent image \mathbf{x} . NNM has an analytical solution and can be solved by soft-thresholding of singular values. In case of NNM, soft-thresholding is performed using single threshold value. However, this assumption is not reasonable as different singular values have different importance and therefore, should be treated differently. [Gu et al., 2014] proposed weighted nuclear norm minimization (WNNM) where every single value is thresholded based on their relative importance based on non-local self-similarity.

2.2.2.5 Principal Component Analysis

Principal Component Analysis (PCA) is successfully employed in many other image restoration fields, and it also found its way to image denoising. Initially, [Muresan and Parks, 2003] proposed the principal component analysis (PCA) on sliding image patches. [Zhang et al., 2010] proposed local grouping of the similar pixels in the neighborhood and then apply PCA to remove the noise, this process is repeated by further refining the output of the first stage. [Deledalle et al., 2011] presented the comparison between different patch-based PCA effect on image denoising.

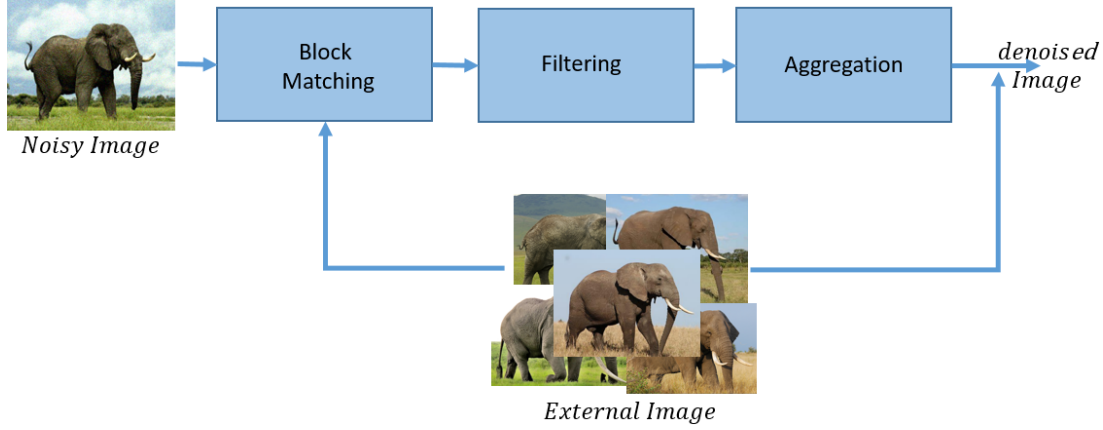


Figure 2.4: A general framework for the external denoising methods.

2.2.2.6 Dual Domain & Progressive Image Denoising

Recently, new algorithms are proposed which operates at pixel level rather than the patches-level. Example of such algorithm is Dual Domain image denoising (DDID) [Knaus and Zwicker, 2013]. It is a simple algorithm which iterates between the spatial domain and the frequency domain using bilateral filtering and short time Fourier transform, respectively. DDID is a guided method which uses the intermediate denoised image as an input to the final stage, although all the stages are identical except the intermediate image. The results are comparative to BM3D besides its simplicity. Furthermore, Progressive image denoising (PID) [Knaus and Zwicker, 2014] is a variant of DDID and improved by incorporating arbitrary fine time steps in iterations. Similar to DDID, it also performs denoising using kernels in two domains: spatial domain and frequency domain. The visual performance of PID is better than DDID.

2.2.3 External Denoising Methods

The use of external image datasets for denoising has been studied in recent years. This trend is motivated by several studies [Levin and Nadler, 2011; Levin et al., 2012] that show that the theoretical minimal error can be achieved by using large datasets. Furthermore, this approach can be made practical by applying efficient sampling techniques on large databases [Chan et al., 2014]. A general framework for the external denoising methods is shown in Figure 2.4.

Denoising of images with known classes is instrumental in various applications such as face image enhancement thus all image solutions tasks where face images are used, document image recovery, digital heritage, cell image analysis, and image

aesthetics to count a few.

Nevertheless, prior research on learning from (external or internal) images [Elad and Aharon, 2006; Zoran and Weiss, 2011; F. Chen and Yu, 2015] only tackled the problem of denoising for natural images. None of them has considered how to denoise object images of a specific class by incorporating class-specific information from object image datasets. There have been efforts in utilizing class-specific priors for image deblurring [Anwar et al., 2015; Sun et al., 2014a], but these approaches are not directly applicable to denoising.

2.2.3.1 Targeted Image Denoising

To complement internal image information, other works [Luo et al., 2014, 2015] resorted to external targeted datasets for image denoising abbreviated as TID. This strategy improves the denoising in specific situations but requires correlated image datasets, and thus fails when the dataset variation becomes high for the same object. Another problem with these algorithms is they involve an exhaustive search policy, which makes them computationally expensive.

2.2.3.2 Combined Image Denoising

Since internal denoising and external denoising have their own strengths, attempts have been made to combine them for denoising [Mosseri et al., 2013; Yue et al., 2014]. [Mosseri et al., 2013] modified the internal denoising to exploit external natural image patches for textured regions. This method introduced a new metric called “PatchSNR” to differentiate between the smooth and textured image regions. PatchSNR is the Signal-to-Noise-Ratio of a patch. The assumption is that for the textured region the PatchSNR is high; hence, external denoising is applied whereas PatchSNR is low for smooth image areas, thus, can take advantage of internal denoising. Using internal or external denoising to a patch based on PatchSNR improves the performance of current internal denoising algorithms.

[Yue et al., 2014] proposed an ad-hoc denoising method where they imposed a restrictive assumption on the external images, which requires them to contain a significant similarity or overlap with the input image. [Yue et al., 2014] combined internal and external BM3D for denoising, hence, the name combined image denoising (CID). Since they employed SIFT [Lowe, 2004] as a keypoint localization step for image registration, the method only works well if the external images are related to the original noisy image by a rigid transformation. Although it has shown promising results in scenarios where the external images are similar to or the same as the input image, but differs in scales and orientations, it fails to demonstrate such performance

when the external images have different content from the input image, even if they belong to the same category.

The difference between both the methods as mentioned above lies in the application of denoising to the patches. [Yue et al., 2014] apply both internal and external denoising to the same patch and then aggregate the outcome of both the denoised patches whereas [Mosseri et al., 2013] either applies either internal or external denoising to a patch.

2.2.4 Learning Patch Statistics

Several learning methods, such as Expected Patch Log Likelihood (EPLL) [Zoran and Weiss, 2011], patch prior guided internal clustering with low rank (PCLR) [Chen et al., 2015], and Patch Group Prior based Denoising (PGPD) [Xu et al., 2015b] were proposed to derive priors from natural noise-free images. However, these learned priors are generic for natural images and are not specific to any image category. Similarly, many early works [Elad and Aharon, 2006; Mairal et al., 2009; Dong et al., 2011] learn an over-complete dictionary of image patches from an external noise-free database and impose non-local self-similarity through a sparse representation. Below are some of the prominent learning methods.

2.2.4.1 Denoising via Singular Value Decomposition

[Elad and Aharon, 2006] proposed a dictionary learning algorithm based on a singular value decomposition (SVD) called K-SVD. It is a simple iterative algorithm and learns from the noisy image itself. Each iteration is composed of two steps: i) learn the coefficients using orthogonal matching pursuit (OMP) [Chen et al., 1989; Pati et al., 1993] (algorithm selects the dictionary atoms sequentially) in each path, ii) update one column of the dictionary at a time. Usually, a few iterations are sufficient to achieve good results. Denoising is performed patch-wise and then inserted back into its original location. Moreover, averaging is conducted in areas of overlapping patches.

2.2.4.2 Non-local Sparse Models

Similar to KSVD, Non-local sparse model (NLSM) also learns from the dictionary from the noisy image. However, the difference is that NLSM applies sparsity on the learned patches. The underlying idea is that similar noisy patches can be approximated using the same sparse decomposition. The purpose of looking for the similar patches are also exploited by many internal denoising algorithms [Dabov et al., 2007b; Buades et al., 2005]. NLSM visual results are considered to be the best

in the current lot such as BM3D, NLM *etc.* However, its computational cost is very high.

2.2.4.3 Spatially Adaptive Iterative Singular Value Thresholding

[Dong et al., 2013] proposed an $\ell_{p,q}$ norm constraint to promote patch similarity and derived a denoising solution via spatially adaptive iterative singular-value thresholding (SAIST).

$$\underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{U}\mathbf{A} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{A}\|_{p,q}, \quad (2.42)$$

where \mathbf{U} is the learned dictionary and \mathbf{A} are the collection of sparse coefficients and $\mathbf{U}\mathbf{A}$ represents the clean image *i.e.* $\mathbf{x}=\mathbf{U}\mathbf{A}$. The $\|\mathbf{A}\|_{p,q}$ is defined by [Cotter et al., 2005] as

$$\|\mathbf{A}\|_{p,q} = \sum_{i=1}^N \|\gamma_i\|_q^p, \quad (2.43)$$

where γ_i is the i -th row of the matrix \mathbf{A} . This algorithm achieved sparse representation using clustering and is formulated by combining the strengths of dictionary learning and structural clustering. In other words, SAIST employs singular value decomposition to represent image patches sparsely and then iteratively remove noise by thresholding the singular values using BayesShrink [Chang et al., 2000]. This method is not only applicable to denoising but other task as well such as image completion. SAIST is computationally expensive as it requires ten iterations to produce the final denoised image.

2.2.4.4 Gaussian Mixture Model priors

As an alternative, [Zoran and Weiss, 2011] aims to learn a statistical prior of natural image patches, such as the Gaussian Mixture Model (GMM) of natural image patches or patch groups for patch reconstruction in a maximum likelihood framework. Expected Patch Log Likelihood (EPLL) contrasts itself than other denoising algorithms by taking a posteriori approach. This method is already described in section 2.1.2.3 as it is applicable for deblurring and denoising with a small difference in formulation. For denoising, the Equation 2.33 becomes

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_i \log p(\mathbf{P}_i \mathbf{x}). \quad (2.44)$$

EPLL uses half-quadratic splitting optimization which introduces auxiliary variables. The process proceeds with alternation between two phases: i) fixing the image

patches while updating auxiliary variables, ii) keeping the auxiliary variables constant and updating the image patches. According to [Zoran and Weiss, 2011], this process is iterated for four to five times. The performance of EPLL is comparable to BM3D and NLSM.

Recently, several authors adapted [Zoran and Weiss, 2011]’s patch prior representing image-specific and class-specific semantics. Based on a Gaussian Mixture Model (GMM), this generic prior captures statistics of natural patches by performing the Expectation-Maximization (EM) algorithm on a large dataset of clean patches [Zoran and Weiss, 2011; F. Chen and Yu, 2015; Xu et al., 2015a].

[Chen et al., 2015] proposed internal clustering guided by external patches and apply low-rank decomposition and termed it as patch prior guided internal clustering with low rank (PCLR). Low-rank regularization is applied on similar internal patches clustered using global similarity in the noisy image rather than local block matching. Subsequently, the method learns Gaussian Mixture Model (GMM) prior to guide the patch clustering and perform a low-rank subspace learning. Such a grouping based low-rank regularization makes the underlying patch restoration very robust to noise. The performance of PCLR is marginally better than BM3D.

Patch Group Prior based Denoising (PGPD) is introduced by [Xu et al., 2015b], which uses a patch group to denoise the noisy patches. After subtraction of mean from the patch group, it represents the non-local self-similar prior to natural images. Then a GMM is learned from the non-local self-similar patch group extracted from natural images. Lastly, sparse coding is applied to the patch group for efficiency. The denoising results of PGPD is below than most state of the art algorithms; however, its efficacy is comparable to BM3D [Xu et al., 2015b].

[Teodoro et al., 2016] proposed an approach to locally adapt the GMM prior [Zoran and Weiss, 2011] to the class of each patch. This method enables patch-based image enhancement for multiple classes appearing in the same image. The authors employ segmentation to differentiate between different classes present in the image; however, this approach may result in degraded denoised outputs as image segmentation itself is a difficult task especially in the presence of noise.

2.2.4.5 Adaptive Image Denoising

AID is an acronym for Adaptive Image Denoising [Luo et al., 2016]. This work is aimed to adapt the generic patch prior to one that is specific to the patch statistics of the input image. The core of the method is a modified version of the Expectation-Minimization (EM) algorithm on the noisy image or its pre-filtered version with an estimate of the noise. The results of this method are superior to the ones when there is no EM adaptation. Also the quantitative results for the proposed image denoising

algorithm yield better results than some state of the art algorithms mentioned earlier.

2.2.5 Convolutional Neural Networks

The advent of convolutional neural networks (CNN) provides a significant performance boost for image denoising methods [Zhang et al., 2017a,b; Lefkimmiatis, 2016; Burger et al., 2012; Schmidt and Roth, 2014] have also been proposed very recently. We present the details of the convolutional neural networks (CNN) in the following sections.

2.2.5.1 Cascade of Shrinkage Fields

[Schmidt and Roth, 2014] learns a single framework based on unification of random-field based model and half-quadratic optimization. The role of the shrinkage in wavelet image restoration is to attenuate small values towards zero due to the assumption of these values being the product of noise instead of the signal values. The pixel values of the shrinkage mappings are learned discriminatively. These predictions are then chained to form a cascade of shrinkage fields (CSF) of Gaussian conditional random Fields. The CSF algorithm considers the data term to be quadratic and must have a closed-form solution based on discrete Fourier transform (DFT).

2.2.5.2 Trainable Nonlinear Reaction-Diffusion

Similarly, [Chen and Pock, 2017] introduced a deep convolutional neural network for image denoising and adapted field-of-experts [Roth and Black, 2009] prior into CNN framework by incorporating a preset number of inference steps. The image restoration model is called Trainable Nonlinear Reaction-Diffusion (TRND).

TRND algorithm extends conventional nonlinear diffusion model to a highly trainable parametrized linear filters and the influence functions. A loss using a significant amount of data is used to learn the parameters such as the filters and the influence functions. The TRND model differs significantly from the conventional models concerning the learned influence functions and the filters. The network of TRND is shown in Figure 2.5

Undoubtedly, CSF and TNRD have shown improved results over more classical methods; however, the imposed image priors inherently impede their performances, which highly rely on the choice of hyper-parameter settings, extensive fine-tuning and stage-wise training.

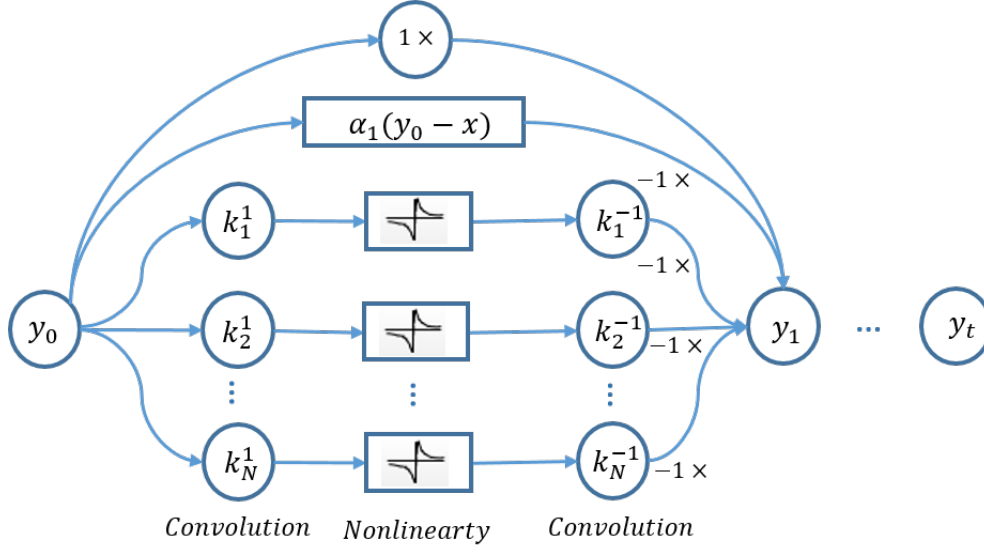


Figure 2.5: The architecture of the TRND network. k_i^1 is the set of linear kernels, y_0 the degraded image, x is the groundtruth image and α_1 is strength of the term.

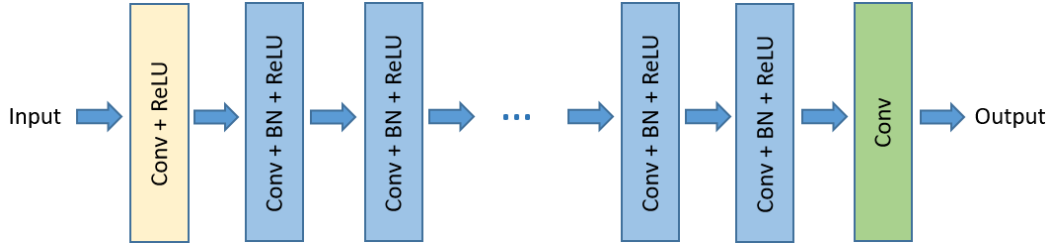


Figure 2.6: The architecture of the DnCNN and IrCNN network.

2.2.5.3 DnCNN & IrCNN

To overcome the drawbacks of CSF and TNRD, IrCNN [Zhang et al., 2017b] and DnCNN [Zhang et al., 2017a] learn the residual present in the contaminated image by using the noise in the loss function instead of the clean image as the ground-truth. The architectures of IrCNN and DnCNN are very simple and similar as it only stacks of convolutional, batch normalization and ReLU layers. The architecture of DnCNN and IrCNN is shown in Figure 2.6.

Although both models were able to report favorable results, their performance depends heavily on the accuracy of noise estimation without knowing the underlying structures and textures present in the image. Besides, they are computationally expensive because of the batch normalization operations after every convolutional layer.

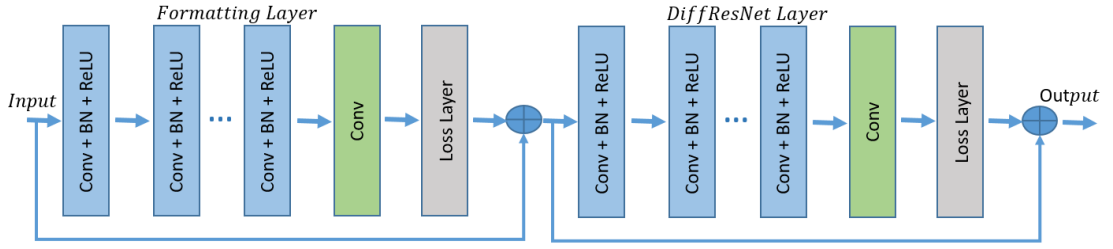


Figure 2.7: FormResNet Proposed network structure.

2.2.5.4 Non-local Color Image Denoising with CNN

Another notable CNN based work is non-local color image denoising abbreviated as NLNet [Lefkimmiatis, 2016] which exploits the non-local self-similarity using deep networks. Non-local variational schemes have motivated the design of the NLNet model and employ the non-local self-similarity property of natural images for denoising. The performance heavily depends on coupling discriminative learning and self-similarity. The restoration performance is comparatively better to several earlier state-of-the-art. Though, this model improves on classical methods but lagging behind IrCNN and DnCNN, as it inherits the limitations associated with the nonlocal self-similarity (NSS) priors as not all patches recur in an image.

2.2.5.5 FormResNet

FormResNet is proposed by [Jiao et al., 2017] which builds upon DnCNN as shown in Figure 2.7. This model is composed of two networks; both networks are similar to DnCNN; however, the difference lies in the loss layers. The first network termed as "Formatting layer" incorporates the Euclidean and perceptual loss. The classical algorithms such as BM3D can also replace this formatting layer. The second deep network "DiffResNet" is similar to DnCNN and input to this network is fed from the first one. The stated formatting layer removes high-frequency corruption in uniform areas, while DiffResNet learns the structured regions. FormResNet improves upon the results of DnCNN by a small margin.

2.2.5.6 Wavelet Domain Deep Network

Recently, a denoising architecture for CNN is proposed by [Bae et al., 2017] to learn the mapping between label datasets to feature space. The motivation behind this CNN based network is persistent homology analysis [Edelsbrunner and Harer, 2008] to that residual learning is a special case of manifold simplification. The network takes the wavelet transforms of images as input and learns the features in the trans-

formed space rather than the original image space. The proposed network has a higher number of channels than DnCNN and FormResNet; hence, the marginal improvement in PSNR can be attributed to the number of channels employed for learning images features.

2.3 Summary

In this chapter, we have presented the essential theories for image deblurring and image denoising and some state-of-the-art solutions to these problems. Image deblurring algorithms can be classified into two groups: 1) blind algorithms, and 2) non-blind algorithms. Image denoising algorithms can be broadly divided into four categories: 1) image filtering, 2) internal denoising, 3) external denoising and 4) learning methods.

The current state-of-the-art internal methods rely on the expertly designed algorithms, and the same is true for learning based methods. The internal methods rely on the internal knowledge heavily. Thus, a question arises, that is it possible to achieve better results while exploiting the external knowledge as compared to the internal one?. Similarly, learning methods capture the statistics of the natural images or patches and require sophisticated algorithms to utilize this knowledge. Here, a question arises, whether is it possible to rely on learning and less on engineering?

Image Deblurring with a Class-Specific Prior

The camera photographs what's there.

Jack Nicholson

A fundamental problem in image deblurring is to recover reliably distinct spatial frequencies that have been suppressed by the blur kernel. To tackle this issue, existing image deblurring techniques often rely on generic image priors such as the sparsity of salient features including image gradients and edges. However, these priors only help recover part of the frequency spectrum, such as the frequencies near the high-end. To this end, we pose the following specific questions: (i) Does class-specific information offer an advantage over existing generic priors for image quality restoration? (ii) If a class-specific prior exists, how should it be encoded into a deblurring framework to recover attenuated image frequencies? Throughout this work, we devise a class-specific prior based on the band-pass filter responses and incorporate it into a deblurring strategy. More specifically, we show that the subspace of band-pass filtered images and their intensity distributions serve as useful priors for recovering image frequencies that are difficult to recover by generic image priors. We demonstrate that our image deblurring framework, when equipped with the above priors, significantly outperforms many state-of-the-art methods using generic image priors or class-specific exemplars.

3.1 Introduction

Image deblurring is an important and long-standing research challenge in low-level vision dating back to 1960s [Trott, 1960]. Blur due to camera shake and camera

motion is still a prevalent issue with images captured by hand-held devices, *e.g.* smartphones or tablet computers. With an exponentially increasing amount of image data captured by these devices, there has been continuing research effort in image deblurring in the last decade [Levin, 2006; Fergus et al., 2006; Shan et al., 2008; Cho and Lee, 2009; Xu and Jia, 2010; Krishnan et al., 2011; Levin et al., 2011a; Xu et al., 2013; Tai et al., 2013; Mosleh et al., 2014; Pan et al., 2014a,b; Xu et al., 2014; Whyte et al., 2014].

In this chapter, we focus our attention on the case of uniform blur, in which a sharp image is convolved with a spatially uniform blur kernel. The goal of blind image deblurring is hence viewed as solving for the latent image \mathbf{x} and the kernel \mathbf{k} given the blurred image \mathbf{y} . By nature, image deblurring is an ill-posed problem, as there exists an infinite number of pairs of latent image \mathbf{x} and kernel \mathbf{k} that result in the same observation \mathbf{y} .

To resolve the above ambiguity, previous works have exploited the sparsity of natural image gradients to impose additional constraints on the deblurring problem. This sparsity constraint is commonly stated in terms of the hyper-Laplacian prior [Krishnan and Fergus, 2009; Levin et al., 2011b], the ℓ_0 [Xu et al., 2013], ℓ_1 [Xu and Jia, 2010] and ℓ_2 -norms [Cho and Lee, 2009], the ℓ_1/ℓ_2 prior [Krishnan et al., 2011], a Gaussian [Levin et al., 2007] or a mixture of Gaussians [Fergus et al., 2006] of the image gradients. A common feature in these works is the presence of a regulariser that minimizes the sparsity of the image gradient. As a result, these methods favor images with strong high-frequency components while ignoring other spatial frequencies. For this reason, these methods are not suitable for many object categories with gradual changes in the surface orientation such as faces, animals, cars, etc.

Furthermore, a common symptom of deblurred images is the presence of ringing artifacts. [Mosleh et al., 2014] has proposed a solution to the detection and removal of ringing by generating a set of Gabor filters that reveals existing ringing artifacts in deblurred images and incorporating these filters in a regularisation scheme to suppress the artifacts. Meanwhile, [Whyte et al., 2014] address the issue of ringing reduction in the presence of saturated pixels. We draw a general remark, from the analysis of these works, that the leading cause of ringing is the suppression of some spatial frequencies by the blur kernel. The frequencies missing from the blur kernel usually cause the Fourier sum of the remaining waves to overshoot at jumps in image intensity. This is known as the Gibbs phenomenon [Hewitt and Hewitt, 1979], rendering ringing artifacts to appear near strong edges.

To overcome the above problem, we leverage prior knowledge of the distribution of frequency components specific to each image class, rather than generic gradient

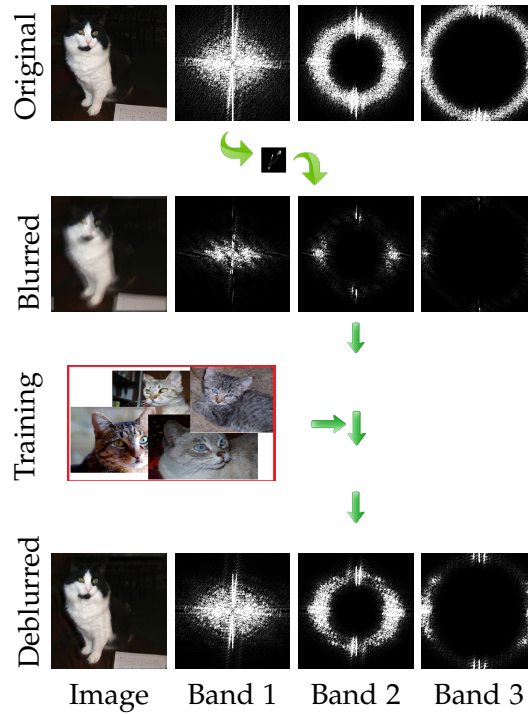


Figure 3.1: Recovering spatial frequencies that have been suppressed by a blur kernel using band-pass frequency components from the training data.

sparsity priors. As a natural choice, we analyze images in the Fourier space due to the convenient transformation of the blur model between the spatial and transfer domain. Instead of imposing a general sparsity constraint, we focus on modeling a class-specific prior in each band of the Fourier spectrum. Specifically, we learn a subspace spanned by the filter responses of sharp images in each class to a band-pass filter. Repeating this learning process over multiple band-pass filters, we capture the characteristics of the target image class across a wide range of frequency bands. The spirit of this work is to discover a more comprehensive prior than those based exclusively on edges or high-frequency image gradients. With our learned priors in hand, we perform the deblurring process in a content-aware fashion.

Figure 3.1 depicts our approach to the restoration of the spatial frequencies attenuated by a blurring kernel. In the first row, we display a sample image from the Cat dataset [Zhang et al., 2008] (first column) and the magnitudes of its Fourier components in three frequency bands (the subsequent columns). A convolution of the input image obtains the frequency components in each band with a Butterworth band-pass filter. Although most of the frequency components of the blurred image (second row) have been annihilated, the recovered image (shown in the last row) contains many frequency components present in the original (in the third row).

Our method is inspired by previous works on image categorization using image statistics. In [Torralba and Oliva, 2003], the authors investigated the spectral signature, i.e., the power spectra of the horizontal and vertical image gradients for each image category. The shape of this spectral signature is an indicator of the scale (size) of the primary element in the scene. This study revealed significant variations in the power spectrum across different image categories, which could enable the categorization of natural and man-made images. In related work, [Geusebroek and Smeulders, 2005] modeled the spatial statistics using a parametric Weibull distribution for the characterization of uniform stochastic textures. Building on this model, subsequent works have proposed methods for image categorization using local texture descriptors [van Gemert et al., 2006]. Specific to image deblurring, [Levin, 2007] integrated the statistics of derivative filters into a maximum likelihood method for blind motion deblurring. However, this study was limited to blurs caused only by a one-dimensional box kernel. The other practical limitation is that it requires the segmentation of the image into layers with common blurs.

We advance the above formulation of image statistics for the purpose of image characterization. In the previous works, image statistics constitute the power spectra of image gradients or derivative filters, which can be viewed as responses to high-frequency filters. In our work, we generalize this notion and consider the distribution of image responses to band-pass filters across all the bands in the frequency spectrum. The novel class prior is based on the following conjectures. Firstly, for every image band, the distribution of band-pass filter responses is characteristic of the image class. Secondly, the band-pass filter responses of images in the same class span a linear subspace. As we shall demonstrate later, these two underpinning conjectures alone are proven to be effective in recovering frequencies suppressed by blur kernels.

To perform blind deblurring, we incorporate the linear subspaces of band-pass filter responses as a class-specific image prior, together with a common ℓ_2 -norm kernel prior into a joint objective function. Subsequently, we employ an iterative optimization approach over several coarse-to-fine image resolutions. In each iteration, the latent image and kernel can be alternately computed as a closed-form solution.

We provide a visual illustration of the relevant training images and bandpass filters selected by the proposed image prior. Figure 3.2 shows an example blurry image (in the second column of the first row), and the four most relevant filtered training images in the second row, together with their weights (shown in the inset). The corresponding training images are displayed in the third row, and the associated bandpass filters are in the fourth row.

It can be seen that the algorithm selects a variety of frequency components from different training images to compose the latent image, including low-frequency de-

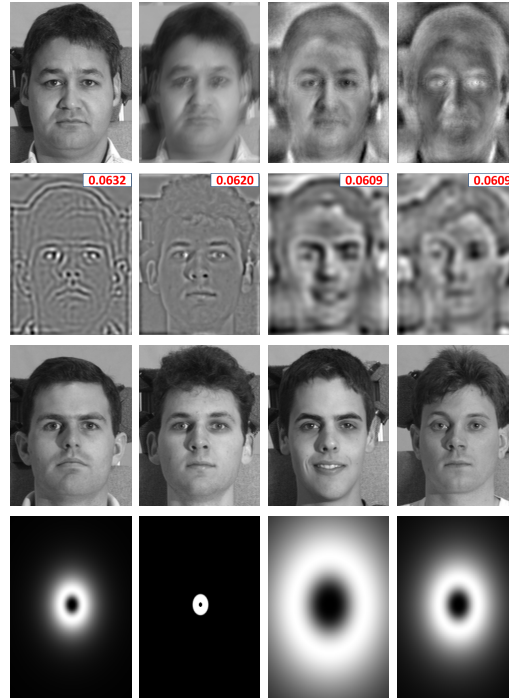


Figure 3.2: A visual demonstration of the proposed prior. Top row (from left to right): original (ground-truth) image \mathbf{x}^* , input blurred image \mathbf{y} , the image reconstructed by the weighted combination of all the filtered training images, and the absolute difference $\|\mathbf{x} - \mathbf{x}^*\|$. Second row (from left to right): the four most important filtered training images sorted by the descending order of their weights (shown in the inset). Third row: the training images corresponding to those in the second row. Fourth row: the bandpass filters (shown in the frequency domain) involved in the filtered training images in the second row.

tails from the first two training images, and mid-frequency details from the latter two. Noticeably, the latent image constructed from the combination of the bandpass components of the training images (in the third column of the first row) is free of blur, especially near edges. Most of the mid-frequency to high-frequency components have been recovered. The absolute difference image (in the fourth column of the first row) shows merely low to mid-frequency details, which could be retrieved by a final non-blind deconvolution step.

The remainder of this chapter is organized as follows. In section 3.2, we formulate the image deblurring problem by incorporating novel class-specific priors. While section 3.3 presents an optimization approach and the closed-form solution to the problem above. Subsequently, in section 3.4, we present results of our deblurring method in a detailed comparison with many previous schemes and also present ablation studies of our algorithm. Lastly, we conclude this work in Section 3.5.

3.2 Problem Formulation

In this section, we introduce our class-specific image priors and derive an optimization problem incorporating these priors. Our problem is stated as follows. Given a set of N sharp training images $\{\mathbf{z}_i | i = 1, \dots, N\}$ and an arbitrary blurred image \mathbf{y} that belongs to the same class, we aim to recover the latent image \mathbf{x} and the kernel \mathbf{k} .

3.2.1 Image Prior

Now we formulate the class-specific image prior, which states that the frequency components in each band span a sparse linear subspace in the Fourier domain. We let $\mathcal{F}_{\mathbf{x}}(\omega)$ denote the Fourier coefficient of the 2D image \mathbf{x} at the spatial frequency ω .

To formulate the problem in the Fourier domain, we obtain a bank of Butterworth bandpass filters, each of which has a constant magnitude in a certain (2D) frequency band and zero elsewhere. The visual representation of bandpass filters in the frequency domain are concentric circular bands (centered at the origin) with unit values. To filter an image with a Butterworth bandpass filter, we first clip its frequency components in the Fourier domain to the range defined by the bandpass filter (corresponding to the filter's non-zero frequencies). Subsequently, the remaining frequency components is transformed to the spatial domain via an inverse Fourier transform.

Having divided the frequency spectrum into a set of M frequency bands, we formulate the linear subspace constraint for band b_j as

$$\mathcal{F}_{\mathbf{x}}(\omega) = \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i}(\omega), \forall \omega \in b_j, j = 1, \dots, M, \quad (3.1)$$

where $w_{i,j}$ is a weight associated with the training image \mathbf{z}_i and the band b_j in the representation of the latent image \mathbf{x} . This coefficient correlates to the similarity between the frequency components of the training and the latent image in the band b_j .

In addition, we enforce sparsity on the weight vector $\mathbf{w}_j \triangleq [w_{1,j}, \dots, w_{N,j}]$ for each band b_j . The sparsity constraint emphasizes the major contributions from a few training images to the representation of the latent image \mathbf{x} for each separate band. Here, we express this constraint as a minimization of the L_1 -norm $\|\mathbf{w}_j\|_1$ due to its well-known robustness.

Combining the linear subspace constraint and the sparsity constraint on \mathbf{w}_j over

all the frequency bands, we define the prior function $P(\mathbf{x}, \mathbf{w})$ as

$$P(\mathbf{x}, \mathbf{w}) \triangleq \gamma \sum_{j=1}^M \sum_{\omega \in b_j} |\mathcal{F}_{\mathbf{x}}(\omega) - \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i}(\omega)|^2 + \tau \sum_{j=1}^M \|\mathbf{w}_j\|_1, \quad (3.2)$$

where γ and τ are the balance factors of the reconstruction error and the sparsity term, respectively, and $|\cdot|$ denotes the modulus of a complex number.

For each band b_j , we define a corresponding band-pass filter \mathbf{f}_j , such as a Butterworth filter [Gonzalez and Woods, 1992], whose Fourier transform is a non-zero constant c within b_j and zero elsewhere. With this filter, let us consider the 2D function $\mathbf{g} = \mathbf{x} * \mathbf{f}_j - \sum_{i=1}^N w_{i,j}(\mathbf{z}_i * \mathbf{f}_j)$, where $*$ denotes the convolution operator. The Fourier transform of this function is

$$\mathcal{F}_{\mathbf{g}}(\omega) = \begin{cases} c \left(\mathcal{F}_{\mathbf{x}}(\omega) - \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i}(\omega) \right) & \forall \omega \in b_j, \\ 0 & \text{otherwise.} \end{cases} \quad (3.3)$$

Applying the Parseval's theorem to the function \mathbf{g} , we have $\int \mathbf{g}(u)^2 du = \int |\mathcal{F}_{\mathbf{g}}(\omega)|^2 d\omega$. Noting that $\int |\mathcal{F}_{\mathbf{g}}(\omega)|^2 d\omega$ is a multiple of the reconstruction error in Equation 3.2, we rewrite it as follows

$$P(\mathbf{x}, \mathbf{w}) = \beta \sum_{j=1}^M \|\mathbf{x} * \mathbf{f}_j - \sum_{i=1}^N w_{i,j}(\mathbf{z}_i * \mathbf{f}_j)\|_2^2 + \tau \sum_{j=1}^M \|\mathbf{w}_j\|_1, \quad (3.4)$$

where we use the variable substitution $\beta \triangleq \frac{\gamma}{c^2}$.

3.2.2 Objective Function

In image deblurring, the aim is to minimize the data fidelity term associated with the blur model $\mathbf{y} = \mathbf{x} * \mathbf{k} + \mathbf{n}$, where \mathbf{n} is the image noise. In addition, several deblurring approaches have utilized $\|\mathbf{k} - \nabla_d \mathbf{y}\|_2^2$, where ∇_d denotes the gradient operator in the direction $d \in \{x, y\}$.

In addition, we employ a regularize on the blur kernel using the conventional L_2 -norm $\|\mathbf{k}\|_2^2$ as in previous works [Cho and Lee, 2009; Yuan et al., 2007]. Combining all the above components, we arrive at minimization of the objective function

$$J(\mathbf{x}, \mathbf{w}, \mathbf{k}) = \|\mathbf{x} * \mathbf{k} - \mathbf{y}\|_2^2 + P(\mathbf{x}, \mathbf{w}) + \sum_{d \in \{x, y\}} \|\nabla_d \mathbf{x} * \mathbf{k} - \nabla_d \mathbf{y}\|_2^2 + \alpha \|\mathbf{k}\|_2^2, \quad (3.5)$$

where α is the balancing factor for the kernel regularizer.

3.3 Deblurring Framework

Given \mathbf{y} , $\{\mathbf{f}_b | b = 1, \dots, M\}$ and $\{\mathbf{z}_i | i = 1, \dots, N\}$, we aim to minimize the objective function in Equation 3.5 with respect to the unknowns \mathbf{x} , \mathbf{w} and \mathbf{k} . Since a simultaneous minimization with respect to all the variables is computationally expensive, we adopt an alternating minimization scheme. In each iteration of this scheme, we solve a sub-problem with respect to one of the variables \mathbf{x} , \mathbf{w} and \mathbf{k} , while fixing the others. The following subsections describe the solution to each sub-problem.

3.3.1 Estimating \mathbf{w} given \mathbf{x} and \mathbf{k}

Assuming that \mathbf{x} , and \mathbf{k} have been obtained in an earlier iteration, we aim to minimize the objective function $J(\mathbf{x}, \mathbf{w}, \mathbf{k})$ with respect to the weights $w_{i,j}$. Here, we note that $P(\mathbf{x}, \mathbf{w})$ in Equation 3.5 can be decomposed into separate bands. Therefore, we can break down the above problem into the minimization of the following function (with respect to \mathbf{w}_j) for each band b_j

$$J_{\mathbf{w}_j} = \|\mathbf{x} * \mathbf{f}_j - \sum_{i=1}^N w_{i,j}(\mathbf{z}_i * \mathbf{f}_j)\|_2^2 + \frac{\tau}{\beta} \|\mathbf{w}_j\|_1. \quad (3.6)$$

We vectorize the images involved in the above Equation using the following shorthand notation $\tilde{\mathbf{x}}_j = \text{vec}(\mathbf{x} * \mathbf{f}_j)$ and $\tilde{\mathbf{z}}_{i,j} = \text{vec}(\mathbf{z}_i * \mathbf{f}_j)$. The minimization of the above cost function can be regarded as an ℓ_1 -regularized least-squares problem and can be solved by standard techniques such as the one reported in [Kim et al., 2007]. The above problem is usually well-formed when the length of $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{z}}_{i,j}$ exceeds that of \mathbf{w}_j , i.e. the number of image pixels is more than the number of training images N .

3.3.2 Latent Image Estimation

With the current update of the contributions \mathbf{w}_j , $j = 1, \dots, M$, from the training images to each band, and the kernel \mathbf{k} , we now estimate the latent image so as to minimize Equation 3.5. Similar to the approach above, we only consider the sum of the terms dependent on \mathbf{x}

$$\begin{aligned} J_{\mathbf{x}} = & \|\mathbf{x} * \mathbf{k} - \mathbf{y}\|_2^2 + \sum_{d \in \{x,y\}} \|\nabla_d \mathbf{x} * \mathbf{k} - \nabla_d \mathbf{y}\|_2^2 \\ & + \beta \sum_{j=1}^M \|\mathbf{x} * \mathbf{f}_j - \sum_{i=1}^N w_{i,j}(\mathbf{z}_i * \mathbf{f}_j)\|_2^2. \end{aligned} \quad (3.7)$$

To this end, we apply the Parseval's theorem to the terms on the right-hand side of Equation 3.7. This theorem states that the total energy of a function over the spatial domain is equal to that of its Fourier transform over the frequency domain. We also note that the image derivative $\nabla_d \mathbf{x}$ can be expressed as a convolution as $\nabla_d * \mathbf{x}$, where ∇_d is a convolution kernel representing the corresponding derivative operation. With these ingredients, we rewrite Equation 3.7 in the Fourier transforms of its terms as

$$\begin{aligned}
 J_{\mathbf{x}} = & \int |\mathcal{F}_{\mathbf{x}}(\omega) \mathcal{F}_{\mathbf{k}}(\omega) - \mathcal{F}_{\mathbf{y}}(\omega)|^2 d\omega \\
 & + \sum_{d \in \{x, y\}} \int |\mathcal{F}_{\nabla_d}(\omega) \mathcal{F}_{\mathbf{x}}(\omega) \mathcal{F}_{\mathbf{k}}(\omega) - \mathcal{F}_{\nabla_d}(\omega) \mathcal{F}_{\mathbf{y}}(\omega)|^2 d\omega \\
 & + \beta \sum_{j=1}^M \int |\mathcal{F}_{\mathbf{x}}(\omega) \mathcal{F}_{\mathbf{f}_j}(\omega) - \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i}(\omega) \mathcal{F}_{\mathbf{f}_j}(\omega)|^2 d\omega,
 \end{aligned} \tag{3.8}$$

where ω represents a spatial frequency, $|\cdot|$ signifies the modulus of a complex number and all the integrals are taken over the entire frequency spectrum.

The Parseval's theorem yields a convenient expression with respect to the Fourier transform of the latent image. Since the function in Equation 3.8 is a convex function of $\mathcal{F}_{\mathbf{x}}(\omega)$ in the Fourier domain, a local optimization method can be applied to obtain its global minimum. Also, we note that $\frac{\partial(|z|^2)}{\partial z} = \bar{z}$, where \bar{z} is the conjugate of the complex number z . For brevity, we omit the frequency ω from the following expressions. By the chain rule, we derive the partial derivative with respect to the Fourier transform $\mathcal{F}_{\mathbf{x}}$ as follows

$$\begin{aligned}
 \frac{\partial J_{\mathbf{x}}}{\partial \mathcal{F}_{\mathbf{x}}} = & 2(\mathcal{F}_{\mathbf{k}} (\overline{\mathcal{F}_{\mathbf{x}} \mathcal{F}_{\mathbf{k}}} - \overline{\mathcal{F}_{\mathbf{y}}}) \\
 & + \sum_{d \in \{x, y\}} \mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{k}} (\overline{\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}} \mathcal{F}_{\mathbf{k}}} - \overline{\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{y}}}) \\
 & + \beta \sum_{j=1}^M \mathcal{F}_{\mathbf{f}_j} (\overline{\mathcal{F}_{\mathbf{x}} \mathcal{F}_{\mathbf{f}_j}} - \sum_{i=1}^N w_{i,j} \overline{\mathcal{F}_{\mathbf{z}_i} \mathcal{F}_{\mathbf{f}_j}})),
 \end{aligned} \tag{3.9}$$

where the multiplications on the right-hand side are performed frequency-wise in the Fourier domain.

We rewrite the complex conjugate of $\frac{\partial J_{\mathbf{x}}}{\partial \mathcal{F}_{\mathbf{x}}}$ as follows

$$\begin{aligned} \overline{\left(\frac{\partial J_{\mathbf{x}}}{\partial \mathcal{F}_{\mathbf{x}}}\right)} &= 2(|\mathcal{F}_{\mathbf{k}}|^2 \mathcal{F}_{\mathbf{x}} - \overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}} \\ &\quad + \sum_{d \in \{x, y\}} (|\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{k}}|^2 \mathcal{F}_{\mathbf{x}} - |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}}) \\ &\quad + \beta \sum_{j=1}^M |\mathcal{F}_{\mathbf{f}_j}|^2 (\mathcal{F}_{\mathbf{x}} - \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i})). \end{aligned} \quad (3.10)$$

By equating the complex conjugate of $\frac{\partial J_{\mathbf{x}}}{\partial \mathcal{F}_{\mathbf{x}}}$ to zero, we obtain the following closed-form solution for the latent image \mathbf{x}

$$\begin{aligned} \mathcal{F}_{\mathbf{x}} &= (\overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}} + \sum_d |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}} + \beta \sum_{j=1}^M |\mathcal{F}_{\mathbf{f}_j}|^2 \sum_{i=1}^N w_{i,j} \mathcal{F}_{\mathbf{z}_i}) ./ \\ &\quad (|\mathcal{F}_{\mathbf{k}}|^2 + \sum_d |\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{k}}|^2 + \beta \sum_{j=1}^M |\mathcal{F}_{\mathbf{f}_j}|^2), \end{aligned} \quad (3.11)$$

where the $./$ notation stands for a frequency-wise division in the Fourier domain. The latent image can be obtained by an inverse Fourier transform of the solution to $\mathcal{F}_{\mathbf{x}}$.

3.3.3 Blur Kernel Estimation

Once the latent image \mathbf{x} is computed, the next step is to estimate the blur kernel \mathbf{k} . Based on Equation 3.5, this optimization step involves the following terms

$$J_{\mathbf{k}} = \|\mathbf{x} * \mathbf{k} - \mathbf{y}\|_2^2 + \sum_d \|\nabla_d \mathbf{x} * \mathbf{k} - \nabla_d \mathbf{y}\|_2^2 + \alpha \|\mathbf{k}\|_2^2. \quad (3.12)$$

Again, we leverage the Parseval's theorem and express the above function in the Fourier domain as

$$\begin{aligned} J_{\mathbf{k}} &= \int |\mathcal{F}_{\mathbf{x}}(\omega) \mathcal{F}_{\mathbf{k}}(\omega) - \mathcal{F}_{\mathbf{y}}(\omega)|^2 d\omega + \alpha \int |\mathcal{F}_{\mathbf{k}}(\omega)|^2 d\omega \\ &\quad + \sum_{d \in \{x, y\}} \int |\mathcal{F}_{\nabla_d}(\omega) \mathcal{F}_{\mathbf{x}}(\omega) \mathcal{F}_{\mathbf{k}}(\omega) - \mathcal{F}_{\nabla_d}(\omega) \mathcal{F}_{\mathbf{y}}(\omega)|^2 d\omega, \end{aligned} \quad (3.13)$$

where, as before, the integrals are taken over the entire frequency spectrum.

Since $J_{\mathbf{k}}$ is a quadratic function of $\mathcal{F}_{\mathbf{k}}(\omega)$, we can obtain the minimiser by setting

$\frac{\partial J_{\mathbf{k}}}{\partial \overline{\mathcal{F}}_{\mathbf{k}}}$ to zero. This derivative can be expanded as

$$\begin{aligned} \frac{\partial J_{\mathbf{k}}}{\partial \overline{\mathcal{F}}_{\mathbf{k}}} &= \mathcal{F}_{\mathbf{x}} (\overline{\mathcal{F}_{\mathbf{x}} \mathcal{F}_{\mathbf{k}}} - \overline{\mathcal{F}_{\mathbf{y}}}) \\ &+ \sum_{d \in \{x, y\}} \mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}} (\overline{\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}} \mathcal{F}_{\mathbf{k}}} - \overline{\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{y}}}) + \alpha \overline{\mathcal{F}_{\mathbf{k}}}. \end{aligned} \quad (3.14)$$

Setting the complex conjugate of the above equation to zero, we obtain the following closed-form solution for $\mathcal{F}_{\mathbf{k}}$ as

$$\begin{aligned} \mathcal{F}_{\mathbf{k}} &= (\overline{\mathcal{F}_{\mathbf{x}}} \mathcal{F}_{\mathbf{y}} + \sum_d |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{x}}} \mathcal{F}_{\mathbf{y}}) / \\ &(|\mathcal{F}_{\mathbf{x}}|^2 + \sum_d |\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}}|^2 + \alpha). \end{aligned} \quad (3.15)$$

For sparse kernels such as motion kernels, which contain mainly high-frequency components, we choose to follow the practice in [Levin et al., 2011a] and include only the image gradient term in the above Equation as its frequency components are more relevant to the kernel spectrum. In that case, the closed-form solution for \mathbf{k} is simplified as

$$\mathbf{k} = \mathcal{F}^{-1} \left(\frac{\sum_{d \in \{x, y\}} |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{x}}} \mathcal{F}_{\mathbf{y}}}{\sum_{d \in \{x, y\}} |\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}}|^2 + \alpha} \right), \quad (3.16)$$

where $\mathcal{F}^{-1}(\cdot)$ denotes the inverse Fourier transform.

3.3.4 Implementation

Our optimization approach is summarized in Algorithm 1. The algorithm takes, as input, a given blurred image \mathbf{y} , a training set of sharp images \mathbf{z}_i , $i = 1, \dots, N$ and a bank of band-pass filters \mathbf{f}_j , $j = 1, \dots, M$, which together cover the entire frequency spectrum. With this input, it aims to compute the latent image \mathbf{x} and the blur kernel \mathbf{k} .

The algorithm commences with the initialization of the latent image and the kernel to the given blurred image and the Dirac delta function, respectively. Subsequently, it proceeds in an iterative manner. In each iteration, we minimize the objective function with respect to \mathbf{w} , \mathbf{x} and \mathbf{k} in alternating steps, as shown in lines 6, 7 and 9. The update steps for \mathbf{x} and \mathbf{k} are undertaken by fast forward and inverse Fourier transforms according to Equations 3.11 and 3.15. After every iteration \mathbf{k} is centered and normalized. Meanwhile, to solve for \mathbf{w} , we minimize the cost function in Equation 3.6 using the L_1 least-squares solver in [Kim et al., 2007]. The algorithm terminates when the values of \mathbf{x} and \mathbf{k} do not change by pre-determined tolerance thresholds over two successive iterations.

Algorithm 1 Deblurring with the class-specific prior.**Input:**

\mathbf{y} : the given blurred image.
 $\mathbf{z}_i, i = 1, \dots, N$: the class-specific training images.
 $\mathbf{f}_j, j = 1, \dots, M$: a set of band-pass filters covering the frequency spectrum.
 α, β : the weights of the terms in Equation 3.5.
 ρ : the attenuation factor of the class-specific prior.

- 1: $\mathcal{F}_x \leftarrow \mathcal{F}_y$.
- 2: $\mathbf{k} \leftarrow \mathbf{ffi}$ (Dirac delta kernel).
- 3: **while** $\text{size}(\mathbf{k}) \leq \text{max_size}$ **do**
- 4: $\beta \leftarrow \beta_0$.
- 5: **repeat**
- 6: Minimize $J_{\mathbf{w}_j}$ in 3.6 w.r.t. $\mathbf{w}_j, \forall j$, with solver in [Kim et al., 2007].
- 7: Update \mathbf{x} according to Equation 3.11.
- 8: $\beta \leftarrow \rho\beta$.
- 9: Update \mathbf{k} according to Equation 3.16.
- 10: **until** the maximum number of iterations is reached or \mathbf{x} and \mathbf{k} change by an amount below a relative tolerance threshold.
- 11: $\mathbf{k} \leftarrow \text{upsample}(\mathbf{k})$ (Initialization of kernel for the following scale) .
- 12: **end while**
- 13: **return** Latent image \mathbf{x} and blur kernel \mathbf{k} .

To improve the stability of the estimates, we progressively increase the kernel size in a coarse-to-fine scheme. Within a fixed kernel scale, we iterate between the estimation steps with respect to \mathbf{w} , \mathbf{x} and \mathbf{k} until convergence, before expanding the kernel size to the next scale. The initial kernel size is 3×3 , and the expansion factor between two successive scales which we found empirically is $\sqrt{1.6}$.

To initialize the kernel in the next scale, we upsample the kernel estimated in the previous iteration using bicubic interpolation. Since iterations at a finer kernel resolution usually inherit good estimates from those at coarser resolutions before further fine-tuning, we enforce a small number of iterations typically between fifteen and twenty for kernel resolutions of 11×11 and above.

In addition, while we preset the weight α of the kernel regularizer, we adjust the weight β of the class-specific prior incrementally over iterations. The reason for this adjustment is that we initially prefer to obtain as much class information as needed to constrain the space of the latent image. On the other hand, as the iterations proceed, we deliberately decrease the influence of this term so that the estimation is increasingly driven by the data fidelity term. In other words, the resulting latent image and kernel will increasingly gather instance-specific details from the given blurred image, rather than the class prior. This step is taken after the update of \mathbf{x} in every iteration, as shown in line 8.

3.3.5 Extension to Color Images

While Algorithm 1 accepts grayscale images as input, it can be extended to deblur color images in a straightforward manner. This extension assumes that all the color channels have been distorted by the same spatially uniform blur kernel. In this case, the variables \mathbf{w} and \mathbf{x} are defined per color channel $c \in \{R, G, B\}$ as \mathbf{w}_c and \mathbf{x}_c , while the kernel \mathbf{k} is the same all the channels. The objective function is then modified as

$$\begin{aligned} J(\mathbf{x}_c, \mathbf{w}_c, \mathbf{k}) = & \sum_c \left[\beta \sum_{j=1}^M \|\mathbf{x}_c * \mathbf{f}_j - \sum_{i=1}^N w_{i,j,c} (\mathbf{z}_{i,c} * \mathbf{f}_j)\|_2^2 \right. \\ & + \|\mathbf{x}_c * \mathbf{k} - \mathbf{y}_c\|_2^2 + \sum_{d \in \{x,y\}} \|\nabla_d \mathbf{x}_c * \mathbf{k} - \nabla_d \mathbf{y}_c\|_2^2 \\ & \left. + \tau \sum_{j=1}^M \|\mathbf{w}_{j,c}\|_1 \right] + \alpha \|\mathbf{k}\|_2^2. \end{aligned} \quad (3.17)$$

The solution for \mathbf{w}_c can be derived by minimizing the following function per channel

$$\begin{aligned} J(\mathbf{x}_c, \mathbf{w}_c) = & \sum_c \left[\beta \sum_{j=1}^M \|\mathbf{x}_c * \mathbf{f}_j - \sum_{i=1}^N w_{i,j,c} (\mathbf{z}_{i,c} * \mathbf{f}_j)\|_2^2 \right. \\ & \left. + \tau \sum_{j=1}^M \|\mathbf{w}_{j,c}\|_1 \right]. \end{aligned} \quad (3.18)$$

Similarly, the update step for \mathbf{x}_c can be performed for each channel using a similar formula to Equation 3.11 as

$$\begin{aligned} \mathcal{F}_{\mathbf{x}_c} = & (\overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}_c} + \sum_d |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{k}}} \mathcal{F}_{\mathbf{y}_c} + \beta \sum_{j=1}^M |\mathcal{F}_{\mathbf{f}_j}|^2 \sum_{i=1}^N w_{i,j,c} \mathcal{F}_{\mathbf{z}_{i,c}}) ./ \\ & (|\mathcal{F}_{\mathbf{k}}|^2 + \sum_d |\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{k}}|^2 + \beta \sum_{j=1}^M |\mathcal{F}_{\mathbf{f}_j}|^2). \end{aligned} \quad (3.19)$$

Meanwhile, the kernel \mathbf{k} is computed by taking a summation of both the numerator and denominator over the color channels as

$$\begin{aligned} \mathcal{F}_{\mathbf{k}} = & \sum_c \left[(\overline{\mathcal{F}_{\mathbf{x}_c}} \mathcal{F}_{\mathbf{y}_c} + \sum_d |\mathcal{F}_{\nabla_d}|^2 \overline{\mathcal{F}_{\mathbf{x}_c}} \mathcal{F}_{\mathbf{y}_c}) \right] ./ \\ & \sum_c \left[(|\mathcal{F}_{\mathbf{x}_c}|^2 + \sum_d |\mathcal{F}_{\nabla_d} \mathcal{F}_{\mathbf{x}_c}|^2 + \alpha) \right]. \end{aligned} \quad (3.20)$$

3.4 Results and Discussion

In this section, we aim to demonstrate the advantage of incorporating the proposed class-specific prior for the blind deconvolution task. For this purpose, we will provide a detailed performance comparison between our method and a number of state-of-the-art alternatives over several datasets. We commence our analysis on the contribution of the components in our framework to the overall performance. Here, we mainly pay attention to the role of the class-specific prior in our framework. Next, we compare our method to a number of well-known deblurring methods that are not equipped with image class priors, in terms of both quantitative and qualitative results. For completeness, we will illustrate the superiority of our method to existing algorithms that exploit class-specific information or class exemplars, in terms of the visual quality of the results.

3.4.1 Datasets and Experimental Settings

We performed the experimental validation on six datasets including the CMU PIE face dataset [Sim et al., 2002], the car dataset in [Krause et al., 2013], the cat dataset in [Zhang et al., 2008], the ETHZ dataset of shape classes [Ferrari et al., 2010], the Yale-B face database [Georghiades et al., 2001] and the INRIA person dataset [Dalal and Triggs, 2005]. For each dataset, we randomly selected half of the images as training data and between 10 and 15 sharp images from the remaining half as ground-truth test images for deblurring. To generate blurred images from the test images, we employed the eight complex ground-truth blur kernels computed by [Levin et al., 2009] from emulated camera shakes. With this input, we compared our proposed algorithm against the state-of-the-art deblurring algorithms with and without using class exemplars under same conditions. The comparison, as will be shown in the following part of the chapter, is based on both the visual quality of the recovered image and blur kernel, as well as the numerical accuracy of these two. In this chapter, we report the numerical error of the full image and kernel in terms of the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR).

We have implemented our algorithm in MATLAB on an Intel CoreTM i7 machine with 16GB of memory. In all of our experiments, we set the parameters $M = 90$, $\alpha = 10$ and $\tau = 0.01$, initialize β to $\beta_0 = 50$, and decrease the value of β by a factor of $\rho = 1.3$ in every iteration until it reaches the minimal value of 0.01. In other words, the contribution from the training images is reduced as the algorithm proceeds and becomes negligible in the end. In the last few iterations, the image and kernel estimation is mainly driven by the information from the blurred image.

For a fair comparison with prior methods, we strive to use the same non-blind



Figure 3.3: Latent images and kernels recovered by our method without (third column) and with the proposed prior (fourth column).

deblurring method, *i.e.* [Levin et al., 2007], in the final step where source code is available and can be modified. As in our method, [Pan et al., 2014a] and [Levin et al., 2011a] already use [Levin et al., 2007]’s method in their original implementation. We also change the default non-blind deconvolution step in [Fergus et al., 2006] and [Shan et al., 2008] to [Levin et al., 2007]. However, the remaining algorithms in our comparison opt for other non-blind deconvolution methods, which cannot be modified in a straightforward manner. For example, [Sun et al., 2013] use [Zoran and Weiss, 2011] as a final non-blind deblurring step. Meanwhile, [Zhong et al., 2013], [Cho and Lee, 2009] and [Xu et al., 2013] devise their own non-blind deconvolution methods and only provide the binary executables of their algorithms. In addition, [Krishnan et al., 2011] utilized their previous work in [Krishnan and Fergus, 2009].

3.4.2 Ablation Study

We perform extensive ablation studies to validate the efficacy of our approach in various aspects.

3.4.2.1 Effectiveness of the Prior

Using the data and setting described above, we demonstrate the effectiveness of the proposed class-specific prior within our deblurring framework. In Figure 3.3, we compare the visual quality of the recovered latent image and the kernel obtained without the prior (in the third column) and with the prior (in the last column). Evi-

	SSIM		PSNR	
Prior	Without	With	Without	With
Images	0.365	0.754	16.87	25.78
Kernel	0.707	0.855	39.42	42.66

Table 3.1: A comparison of the accuracy achieved by our deblurring framework on all the mentioned datasets with and without the proposed prior.

	SSIM			PSNR		
	Intensity only	Gradient only	Both	Intensity only	Gradient only	Both
Images	0.678	0.529	0.754	23.28	21.16	25.78
Kernel	0.819	0.745	0.855	41.27	40.01	42.66

Table 3.2: Influence of intensity and gradient fidelity terms on the deblurring results.

dently, the image recovered with the prior does not contain visible artifacts, whereas that obtained without the prior shows severe ringing and multiple false edges. Also, when inspecting the estimated kernel (better viewed when zoomed in the electronic copy), we observed a noisy one close to the delta kernel (the initial kernel) when we do not include the image prior in our method. This suggests that the method may not have converged without this prior. On the other hand, the kernel is almost identical to the ground-truth with the prior included.

Further, we have quantified the accuracy of the recovered latent image and blurred kernel with and without the use of the class-specific prior. In Table 3.1, the accuracy is measured in SSIM and PSNR, indicating the similarity between the estimated quantities and the corresponding ground-truth. These results demonstrate that the accuracy of the recovered image and kernel improves significantly (by several orders of magnitude) with the proposed prior. This is consistent with the visual observations above, suggesting that the proposed prior plays an important role in correctly guiding the estimates to the ground-truth.

We have also performed an experiment where the prior only covers the mid and high frequency bands, and the low frequency components of the latent image are estimated directly from the input image. Without the low frequency in the prior, the average PSNR for the CMU dataset declines to 25.67 dB, as compared to 30.75 dB (in Table 4, when all the frequency bands are incorporated in the image prior). Hence, incorporating low frequency bands is actually beneficial, rather than harmful, to the deblurring task.

3.4.2.2 Influence of Data Fidelity Terms on Kernel Estimation

We experimented with different options of the data terms for the estimation of latent image (while only employing the gradient information for estimating the kernel). These includes the intensity fidelity term, *i.e.* $\|\mathbf{x} * \mathbf{k} - \mathbf{y}\|_2^2$, and the gradient fidelity term, *i.e.* $\|\nabla_d \mathbf{x} * \mathbf{k} - \nabla_d \mathbf{y}\|_2^2$ in Equation 3.5, or both.

Table 3.2 shows the accuracy of the deblurred image and kernel estimate across all the datasets under study, in terms of SSIM and PSNR. The highest accuracy is achieved when both data fidelity terms, *i.e.* intensity and gradients, are employed jointly with the class specific prior, while using each individual fidelity term yields a lower accuracy. For this reason, we employ both the intensity and gradient fidelity terms in our framework. Table 3.2 shows the accuracy of the deblurred image and kernel estimate across all the datasets under study, in terms of SSIM and PSNR. The accuracy is lower when data fidelity terms *i.e.* intensity and gradients are used for deblurring individually with class-specific prior, while higher accuracy is achieved when both terms are combined for deblurring. For this reason, we employ both the intensity and gradient fidelity terms in our framework.

Figure 3.4 illustrates example kernels estimated under the above three settings. Specifically, the kernels in Figures 3.4(d)-(f) are recovered using only either the intensity or the gradient fidelity term, and then with both terms, respectively. Among these options, the former two yield kernel estimates with clear structural deviations from the ground-truth shown in Figure 3.4c. On the other hand, the kernel yielded using both fidelity terms (Figure 3.4f) is closer to the ground-truth. This implies a more accurate estimation of the intermediate latent image.

3.4.2.3 Influence of the Dataset Size

We also examine the variation of deblurring performance with respect to the number of training images. Table 3.3 shows that the image and kernel estimation accuracy for the CMU PIE dataset improves consistently with the increasing number of training images. Even with only 50 training samples, our method can achieve an average image accuracy of 25.87 dB, outperforming all the other methods on this dataset (More detailed results are given in Table 3.8).

Training size	50	125	250	500	1000	2000
Image	25.87	26.97	27.92	28.58	30.42	30.75
Kernel	41.15	42.11	42.80	42.99	44.01	44.13

Table 3.3: Deblurring performance (in PSNR) on the CMU PIE dataset for different numbers of training images.

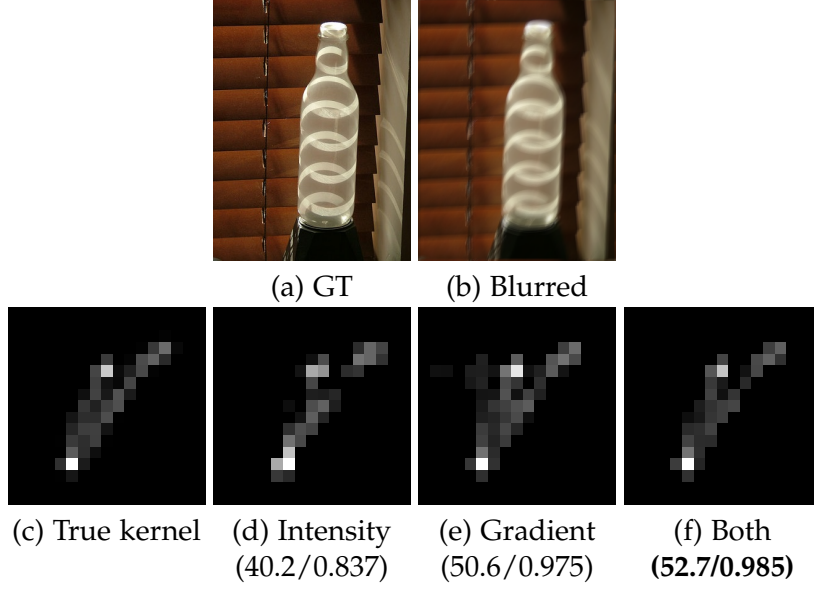


Figure 3.4: Influence of the data fidelity term in the objective function on the kernel estimate. A pair of PSNR/SSIM error metrics is shown for each kernel estimate in the sub-figures (d)–(f). (a) ground-truth image, (b) blurred image, (c) ground-truth kernel, (d) estimated kernel with the intensity term only, (e) estimated kernel with the gradient term only, (f) estimated kernel with both terms.

3.4.2.4 Choice of the Training Class

We ask the question whether the choice of training class significantly alters the deblurring accuracy. To this end, we experiment with various pairs of training and test object categories. Table 3.4 shows the accuracy of deblurred images (in PSNR) for various training (along with the columns) and test (along with the rows) categories. In most cases, the best accuracy is achieved along the diagonal, *i.e.* when the input test image belongs to the same object category as the training dataset. The only exception is that our algorithm achieves the best deblurring accuracy for the Yale-B dataset when being trained on the Cat dataset. This result matches the observation that some features of a cat face such as eyes, lips, and contours resemble those of a human face. As a consequence, employing cat faces as training data are potentially as beneficial for the deblurring of human faces. Otherwise, the PSNR degrades significantly when the training class differs from the test class. These results demonstrate the impact of choosing the correct training class on the deblurring accuracy.

Further, we have evaluated our algorithm with training examples combined from all object classes, and compared its performance to the case of separate training classes. The last column in Table 3.4 reports the image PSNR using training exam-

Test \ Train	Bottles	Car	Cat	CMU	Human	Yale	All
Bottles	23.43	20.11	20.91	20.34	20.41	20.87	22.41
Car	21.65	24.51	21.93	20.75	19.93	20.21	23.56
Cat	20.92	18.31	30.10	21.66	20.88	20.25	25.32
CMU	28.19	27.36	26.68	30.75	26.13	28.15	29.35
Human	14.24	15.07	13.96	12.95	18.56	14.92	18.47
Yale	27.96	25.49	29.24	29.02	25.71	29.04	28.03

Table 3.4: Deblurring performance (in PSNR) for different classes of the blurred input image and the external training datasets. The PSNR is significantly higher when the external dataset matches the input image category.

β	50	5	1	0.5	10^{-1}	10^{-2}	10^{-3}	10^{-4}
PSNR	9.35	8.57	7.57	10.80	16.55	16.90	16.15	16.01

Table 3.5: The average image accuracy (in PSNR) achieved with a constant prior weight β when our algorithm is evaluated on the Person dataset [Dalal and Triggs, 2005].

ples from all object classes. Indeed, including all the object classes in the training data degrades the image accuracy compared to only the correct training class. This result is an evidence that examples within the same class are more beneficial to the deblurring accuracy than those outside the class.

3.4.2.5 Schedule of the Prior Weight β

We have assessed the performance of our algorithm with a fixed weight β over all the iterations. In Table 3.5, we present the accuracy of the latent image (in PSNR) recovered for the INRIA human dataset, with respect to different constant values of β . The image PSNR suffers severely from an overweighted image prior (when $\beta \geq 1$) and varies slightly with a smaller prior weight, *i.e.* no more than 10^{-1} . The highest PSNR of 16.90 dB is observed for $\beta = 10^{-2}$. However, it is worth noting that this level of accuracy is still several orders of magnitude lower than the image PSNR of 18.56 dB, which is reported in the last row and the “Person” column in the PSNR section of Table 3.8. This comparison demonstrates that the strategy of attenuating β by a factor of $\rho = 1.3$ in every iteration is more effective than using a constant prior weight.

3.4.2.6 Number of Bandpass Filters

We also evaluate our algorithm performance with different numbers of bandpass filters *i.e.* M using the same setting for other parameters. In Table 3.6, we observe

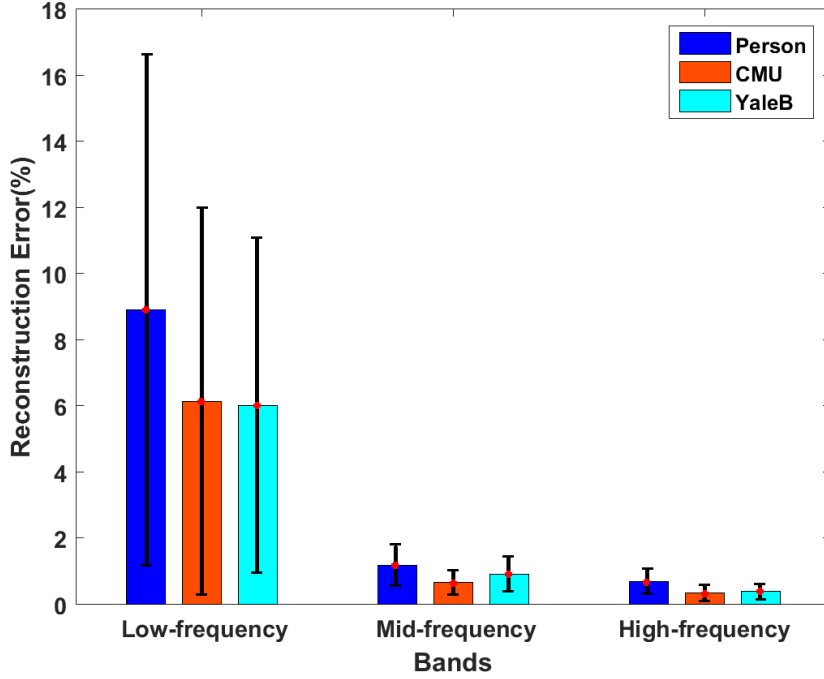


Figure 3.5: The relative reconstruction errors (averaged over 80 test images) for the INRIA person [Dalal and Triggs, 2005], the CMU-PIE [Sim et al., 2002] and the Yale-B [Georghiades et al., 2001] datasets.

that the average image PSNR for the Person dataset [Dalal and Triggs, 2005] varies gradually with respect to different values of M . Since the peak PSNR is achieved at $M = 90$, we employ 90 filters throughout all other experiments.

No. filters	10	30	50	70	90	110	130
PSNR	17.31	17.81	17.94	18.12	18.56	18.52	18.53

Table 3.6: The average accuracy of the deblurred image (in PNSR) for the Person dataset [Dalal and Triggs, 2005], with respect to different numbers of bandpass filters M .

3.4.2.7 Reconstruction Error of the Latent Image

To validate the prior, we report the relative error of the latent image reconstructed by the weighted combination of the filtered training images. We evenly divide the 90 bandpass filters into three groups, corresponding to the low-frequency, the mid-frequency, and the high-frequency bands, and study the reconstruction error per group. Figure 3.5 shows the average relative reconstruction error for 80 blurry images in the INRIA person [Dalal and Triggs, 2005], the CMU-PIE [Sim et al., 2002] and the Yale-B [Georghiades et al., 2001] datasets, across the above three groups of frequency

	80 filters		90 filters	
	Greyscale	Colour	Greyscale	Colour
Images	18.31	18.33	18.56	18.57
Kernel	38.68	39.65	41.32	41.78

Table 3.7: A comparison of the image and kernel accuracy (in PSNR) obtained using greyscale vs. colour input images. The results are reported for the INRIA person dataset [Dalal and Triggs, 2005].

bands. For each input image, we employ 100 training images from the same class.

Overall, the average errors in most cases are reasonably low (6% and below), except the 9% error for the low-frequency bands in the INRIA Person dataset. This could be explained by the fact that this dataset contains a wider variety of human poses and background than the other datasets. In particular, in the mid-frequency and high-frequency regions, the error mean is 1% or below and one standard deviation above the error mean lower than 2%, across the datasets. This supports the claim that, with a sufficient number of training images and bandpass filters, we can recover the mid-frequency and high-frequency details of the blurry images with a high level of accuracy.

3.4.2.8 Grayscale vs. Color

We compare the accuracy of our algorithm when it is run on input color images as opposed to their grayscale counterparts. As a demonstration, we perform this comparison on the INRIA human dataset [Dalal and Triggs, 2005], using $M = 80$ and $M = 90$ bandpass filters. Table 3.7 reports the accuracy (in PSNR) of the kernel estimate and the final deblurred image. Under both settings, the kernel PSNR obtained from color input images is higher than that from the grayscale ones. However, there is no clear correlation between the kernel PSNR and the image PSNR as the latter is almost unaffected by the input modality. The explanation is that, although the kernel estimated from color images is more accurate, it may still lack some frequency components in the original image. Therefore, these components could not be recovered from either the grayscale or the blurry color image directly, but could only be hallucinated using image priors.

3.4.2.9 Convergence

The objective function in Equation 3.5 is convex with respect to each of the variables \mathbf{w} , $\mathcal{F}_x(\omega)$ and $\mathcal{F}_k(\omega)$. When two of these three variables are fixed, the overall objective function is reduced to those in Equations 3.6, 3.8 and 3.13. Those objective

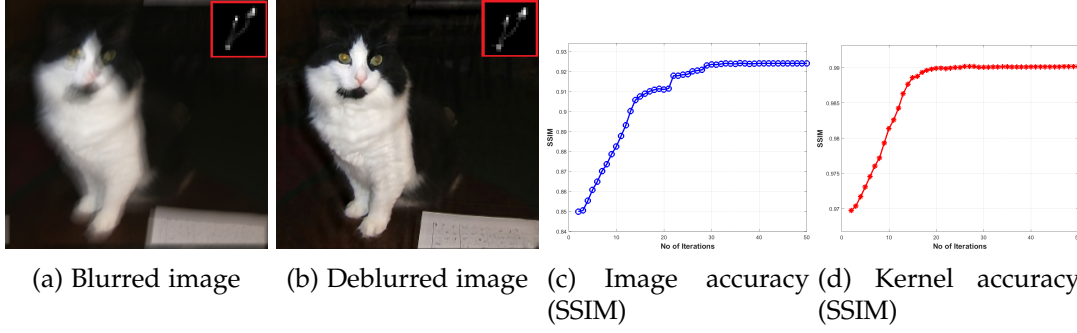


Figure 3.6: The convergence of the iterative algorithm. The image and kernel similarity between the estimated and the ground-truth are measured in terms of the SSIM.



Figure 3.7: Estimated kernels for the sample image in Figure 3.6 at different scales. As visible, the kernel becomes progressively more similar to the ground-truth at finer resolutions.

functions are convex with respect to the respective variables to be optimized, because they consist of a quadratic term, and an additional ℓ_1 regularization term when the weights \mathbf{w} are to be optimized.

Therefore, each alternating minimization step between lines 5 and 10 of Algorithm 1 is guaranteed to converge to a global minimum for each subproblem. Overall, the algorithm converges to a local minimal solution for the variable triplet \mathbf{w} , \mathbf{x} and \mathbf{k} .

In Figure 3.6a, we demonstrate the convergence of our algorithm on a sample image. The top row shows the input (left) and the deblurred image (right). In Figures 3.6c and 3.6d, we plot the similarity of the estimated image and kernel to the corresponding ground-truth with respect to the iteration number on the finest scale. Here, the similarity is measured by SSIM. The overall trend is that the estimated image and kernel become increasingly similar to the ground-truth in the long run, and the image and kernel similarity measure plateau at high values, above 0.92 and 0.99, respectively. In addition, Figure 3.7 illustrates the progression of the estimated kernel from the coarsest to the finest resolution, for the blurred image in Figure 3.6a.

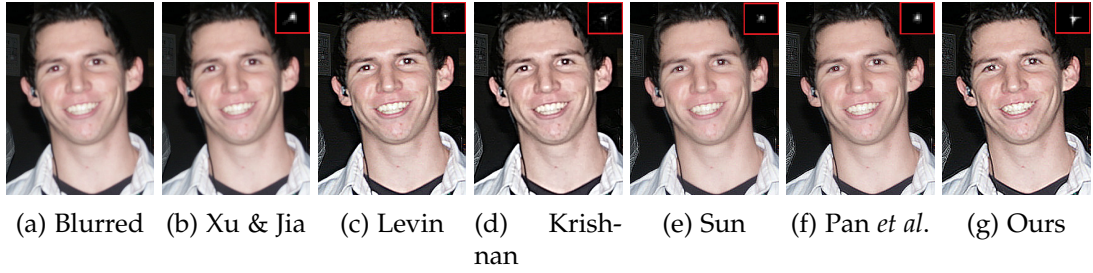


Figure 3.8: Deblurring a real-world image (with no known ground-truth) from the dataset in [Shi et al., 2014].

As shown, the estimated kernel is progressively closer to the ground-truth as the resolution becomes finer.

3.4.2.10 Runtime

One can deduce the complexity of our algorithm in its basic implementation. It is necessary to specify the complexity of each loop, and each step of the algorithm *i.e.* Equation 3.6, 3.11 and 3.16. Let m be the number of pixels of the input image, then the computational load of Equation 3.6 for N training images is Nm . Equation 3.11 requires two 2D Fast Fourier Transforms (FFT) and an inverse FFT in each inner iteration/minimization step. Notice that, the bandpass filters, the derivative filters and responses for the training images are precomputed. Therefore, after simplification and ignoring the constants values, the complexity of Equation 3.11 is $O(m \log m)$. Similarly, for Equation 3.16 one can see the the complexity to be same as Equation 3.11 *i.e.* $O(m \log m)$. The computational complexity of the first pass of the main loop is $O(m \log m + Nm)$, while it takes σ inner iterations and $k_s/3$ outer iterations to give us the final kernel. Hence, the overall complexity of our method is $O(\sigma k_s m \log m + \sigma k_s Nm)$, where k_s is the size of the kernel. The execution time for a 320×240 image is 33 seconds with our MATLAB implementation without any code optimization. Notice that, the weight estimation step is highly parallelizable and the 2D FFT operations typically run in real-time at full framerate for much larger video frames in dedicated hardware platforms.

3.4.2.11 Real-world Images

We also demonstrate how our method performs on real-world examples, where the original sharp images are unavailable. For example, Figure 3.8 shows an example of a real-world blurred face image from the dataset in [Shi et al., 2014]. It is worth noting that, although the dataset in [Shi et al., 2014] contains images induced by spatially non-uniform kernels, we assume that the facial area in the example shown



Figure 3.9: Deblurring results for real input images from [Pan et al., 2014a], where the one in the first row contains noise and saturated pixels.

in Figure 3.8 has been blurred by an approximately uniform kernel, due to the similar depth across the foreground (facial) area. We also simply ignore the details in the dark background. Here, we use a fixed size blur kernel (19×19 pixels) as the input to our algorithm. Our algorithm recovers almost all of the fine facial and hair textures and sharp highlights in the eyes while the state-of-the-art [Sun et al., 2013; Xu and Jia, 2010] produce over-smoothed images. In this example, our results are competitive, if not better, than the state-of-the-art.

Figure 3.9 illustrates the qualitative results for two more such examples from [Pan et al., 2014a]. It is noted that the first example contains noise and saturated pixels. Here, we employ the same kernel size as [Pan et al., 2014a], *i.e.* 35×35 for the first image and 25×25 for the second one. For the first example, our algorithm recovers finer facial details and hair textures and smoother facial skin than the remaining methods, whereas the others produce ringing artifacts and amplify noise. It is demonstrated through this example, that our method can smooth out a certain level of input noise.

In the second example, the methods in [Pan et al., 2014a; Sun et al., 2013; Xu and Jia, 2010] produce blurry images with ringing artifacts, perhaps due to the sub-optimal selection of edge scales for kernel estimation. Our result is competitive to [Krishnan and Fergus, 2009] while yielding finer facial details and less ringing artifacts than [Levin et al., 2011b].

3.4.2.12 Reconstruction of a Cat Image

In Figure 3.10, we illustrate the reconstruction of an example from the Cat dataset. It can be seen that the proposed prior selects a variety of frequency components of

different training images in various poses. Although the reconstructed latent image lacks minute details such as whiskers and furs, most of its content (especially around the edges) is free of blur.

The latent image was reconstructed with not only the shown examples but more than five thousand training images. Within each frequency band, the filtered input image is projected onto this basis to minimize the ℓ_2 reconstruction loss. The *over-complete* dictionary of training examples enables the recovery of most mid-frequency to high-frequency components.

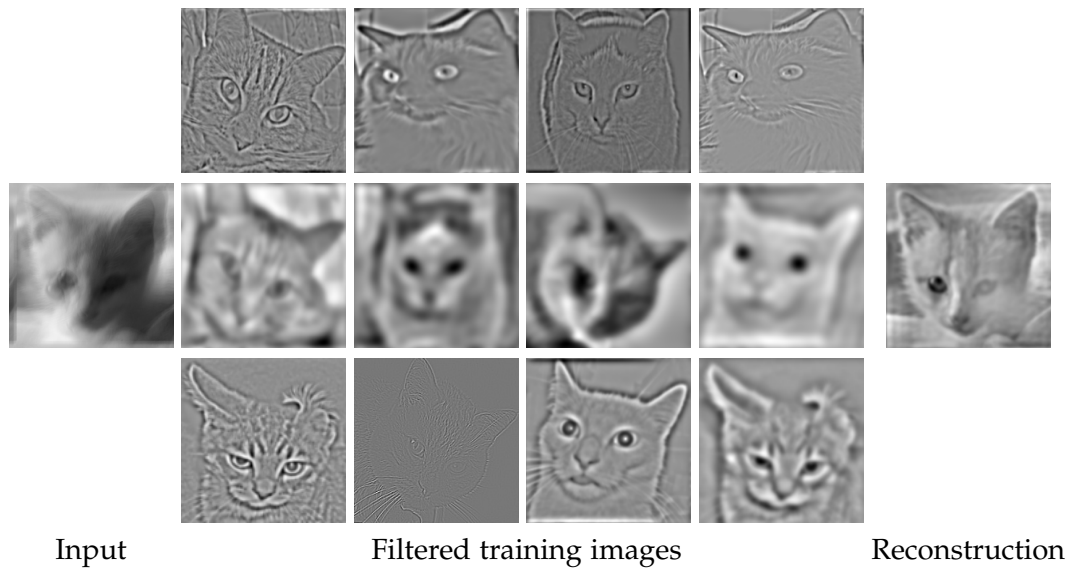


Figure 3.10: The reconstruction of a cat image, by taking the weighted combination of all the filtered training images from the Cat dataset. From left to right: blurred input image, important filtered training images, and reconstructed image.

3.4.2.13 Distribution of weights of filtered training images

We plot a 2D colour map in Figure 3.11 for the weights used for the reconstructed image in Figure 3.2. The columns correspond to the contributing images while the rows correspond to the frequency bands. The bands are divided into low-frequencies (rows 1 to 30), mid-frequencies (row 31 to 60), and high frequencies (row 61 to 90). The filter weights are colour-coded, where red means high weights and blue means low weights. It can be seen from the plot that high-frequency bands contribute the most to the reconstruction, followed by mid and low-frequency bands.

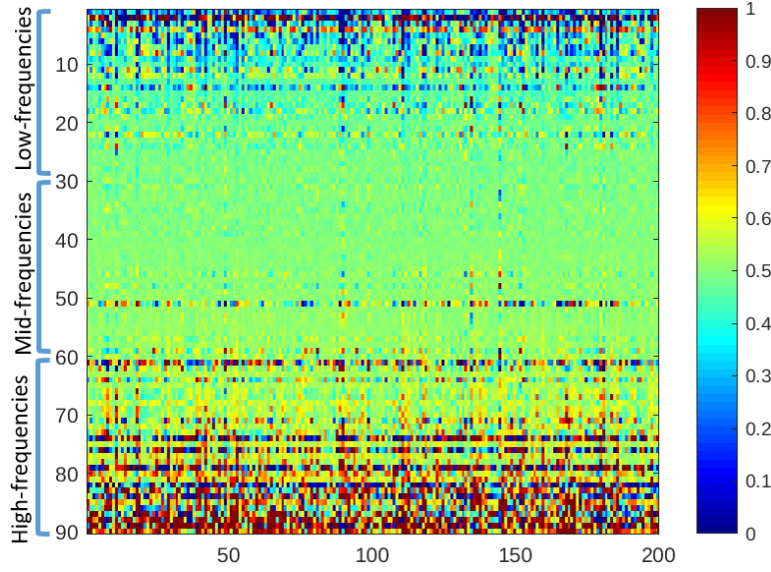


Figure 3.11: The weights $w_{i,j}$ for the reconstruction of the latent image in Figure 3.2.

3.4.3 Comparisons with Generic Image Deblurring

In this section, we evaluate the performance of our method as compared to several state-of-the-art deblurring methods that use generic priors on the datasets mentioned earlier. The methods included in our comparison are that of [Fergus et al., 2006], [Shan et al., 2008], [Cho and Lee, 2009], [Xu and Jia, 2010], [Krishnan et al., 2011], [Levin et al., 2011a], [Cai et al., 2012], [Zhong et al., 2013], [Xu et al., 2013], [Sun et al., 2013] and [Pan et al., 2014a].

In Table 3.8, we present the average SSIM and PSNR scores for the recovered latent images. Among all methods, ours is the best performer across all datasets tested. Our method outperforms the second-best performer by more than 15% in terms of the SSIM scores. Our PSNR results are several dB higher thanks to the ability of our method in capturing frequency-wise details in each image class. This aspect distinguishes our method from the generic approaches, which mainly employ sparse intensity or gradient priors and, as a consequence, favor reconstructions with uniform regions.

We report the SSIM and PSNR scores for the estimated kernels in Table 3.9. Again, our method outperforms all other deblurring algorithms. On all six datasets, the SSIM scores of our method are the best, achieving at least 6.5% higher scores than the state-of-the-art. When the kernel similarity measured by PSNR, our method leads the second best by several orders of magnitude, noting that PSNR is computed in the logarithmic base. Since our method captures class-specific information in every

PSNR (dB)						
Methods	Car	Shape	Cat	CMU	Person	YaleB
[Fergus et al., 2006]	16.99	16.69	19.88	18.26	14.81	19.56
[Shan et al., 2008]	21.56	21.61	25.20	25.59	17.78	26.42
[Cho and Lee, 2009]	19.99	20.47	22.54	24.38	15.05	22.99
[Xu and Jia, 2010]	20.93	21.25	22.73	23.30	-	23.30
[Krishnan et al., 2011]	19.75	19.73	22.79	23.54	15.41	24.09
[Levin et al., 2011a]	18.09	19.24	23.12	24.31	16.77	25.22
[Cai et al., 2012]	13.89	14.86	14.63	11.72	-	12.37
[Zhong et al., 2013]	17.23	18.00	20.73	20.93	-	22.16
[Sun et al., 2013]	19.06	22.50	23.93	24.78	-	23.74
Ours	24.51	23.43	30.10	30.75	18.56	27.35
SSIM						
[Fergus et al., 2006]	0.411	0.415	0.598	0.559	0.207	0.535
[Shan et al., 2008]	0.632	0.624	0.742	0.775	0.407	0.773
[Cho and Lee, 2009]	0.559	0.595	0.627	0.699	0.293	0.678
[Xu and Jia, 2010]	0.631	0.638	0.704	0.739	-	0.681
[Krishnan et al., 2011]	0.544	0.544	0.668	0.693	0.296	0.755
[Levin et al., 2011a]	0.500	0.567	0.699	0.758	0.332	0.673
[Cai et al., 2012]	0.298	0.358	0.292	0.178	-	0.205
[Zhong et al., 2013]	0.485	0.520	0.643	0.641	-	0.655
[Sun et al., 2013]	0.481	0.669	0.724	0.744	-	0.680
Ours	0.765	0.715	0.864	0.881	0.509	0.788

Table 3.8: The accuracy of the deblurred images, measured by SSIM and PSNR. The missing results, indicated by “-”, occurs when the respective method is not capable of dealing with the low resolution of the input images. Best results are in bold.

frequency band of the latent image, it is capable of coping with a broad range of kernels, irrespective of whether they are sparse or not.

For a comprehensive evaluation, we present the qualitative comparisons for sample images from the datasets under study. As the first example, in Figure 3.12, we show the deblurring results for a car image from the dataset in [Krause et al., 2013]. Overall, our method produces the image with the smallest amount of artifacts and the most accurate kernel. Note that the ground-truth image in Figure 3.12a does not contain much texture except for a small number of edges. Therefore, the methods that amplify edges such as those in [Cho and Lee, 2009; Xu and Jia, 2010; Xu et al., 2013; Pan et al., 2014a] receive limited information, thus, cannot handle this case well. Moreover, the methods based on gradient sparsity priors, including those in [Krishnan et al., 2011; Levin et al., 2011a; Cai et al., 2009], tend to produce artifacts in the deblurred image. [Levin et al., 2011a] (Figure 3.12g) appears to generate a

PSNR (dB)						
Methods	Car	Shape	Cat	CMU	Person	YaleB
[Fergus et al., 2006]	37.27	37.80	37.43	40.03	36.56	40.44
[Shan et al., 2008]	41.26	40.92	40.95	41.00	40.34	40.87
[Cho and Lee, 2009]	41.51	41.05	41.25	41.00	40.78	40.92
[Xu and Jia, 2010]	41.39	41.39	41.35	41.39	-	41.09
[Krishnan et al., 2011]	39.10	39.14	39.16	40.38	38.99	40.10
[Levin et al., 2011a]	39.12	38.93	39.28	39.92	37.36	40.13
[Cai et al., 2012]	38.62	38.26	38.90	39.33	-	39.19
[Zhong et al., 2013]	39.41	39.92	40.15	40.41	-	40.16
[Sun et al., 2013]	41.15	41.46	40.62	40.91	-	40.84
Ours	43.78	41.61	43.82	44.14	41.32	41.28
SSIM						
[Fergus et al., 2006]	0.629	0.668	0.653	0.759	0.589	0.778
[Shan et al., 2008]	0.831	0.816	0.819	0.816	0.778	0.780
[Cho and Lee, 2009]	0.845	0.830	0.834	0.820	0.803	0.809
[Xu and Jia, 2010]	0.840	0.761	0.837	0.840	-	0.815
[Krishnan et al., 2011]	0.724	0.721	0.719	0.787	0.698	0.760
[Levin et al., 2011a]	0.702	0.692	0.750	0.782	0.621	0.762
[Cai et al., 2012]	0.640	0.627	0.652	0.669	-	0.662
[Zhong et al., 2013]	0.704	0.755	0.764	0.774	-	0.748
[Sun et al., 2013]	0.829	0.822	0.816	0.826	-	0.813
Ours	0.884	0.833	0.886	0.901	0.805	0.823

Table 3.9: The similarity between the estimated kernel to the ground-truth, measured by SSIM and PSNR. The missing results, indicated by “-”, occurs when the respective method is not capable of dealing with the low resolution of the input images. Best results are in bold.

similar result to our method (Figure 3.12o). However, a close inspection reveals that [Levin et al., 2011a] contains ringing defects in the deblurred image and undesirable non-zeros in the kernel.

As a second example, we present deblurring results on a challenging image from the INRIA person dataset [Dalal and Triggs, 2005]. As shown in Figure 3.13, the input image has a low resolution of 64×80 and incurs severe blur due to the large kernel size relative to the image size. Note that the results for [Sun et al., 2013] and [Xu et al., 2013]’s methods are not available since their original implementations are unable to handle the low image resolution. Since all the edges are significantly distorted, sparsity and gradient priors do not benefit the deblurring task. For this reason, it is difficult for the methods that utilize these priors, such as those in [Cho and Lee, 2009; Xu and Jia, 2010; Zhong et al., 2013], the MAP frameworks [Shan

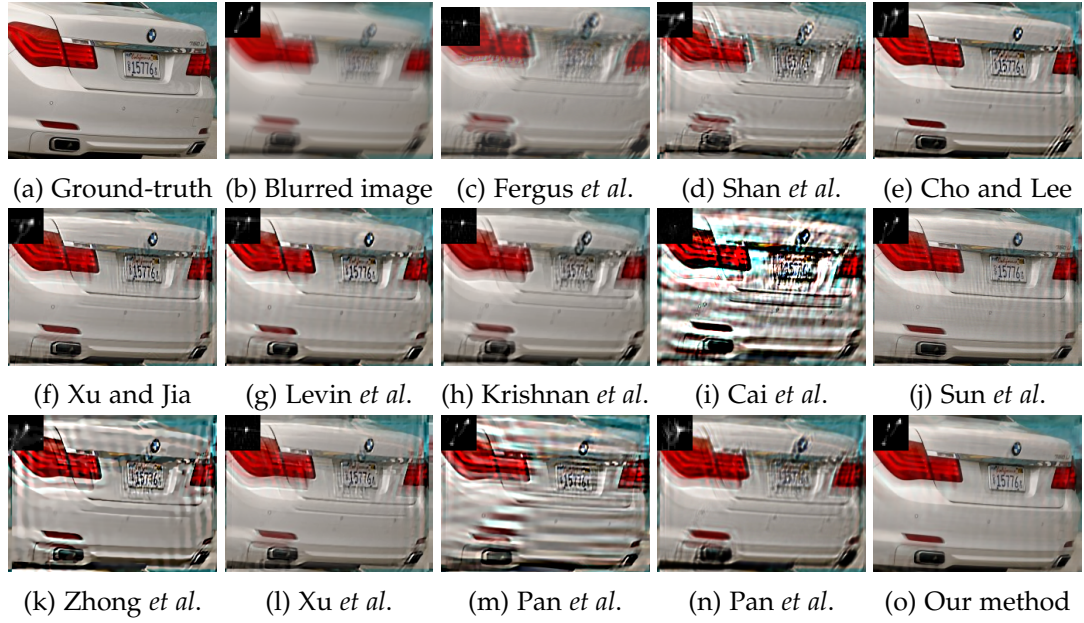


Figure 3.12: Results for a sample image from the Car dataset in [Krause et al., 2013]. The restored image from our method has more legible text on the license plate compared to the other methods.

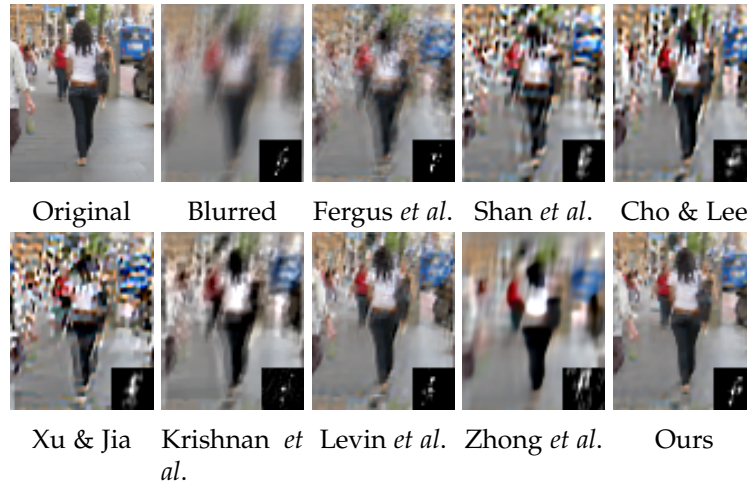


Figure 3.13: Comparison of several methods on a sample image selected from the INRIA dataset [Dalal and Triggs, 2005]. Our method successfully recovers parts of the image with a significant resemblance to the ground-truth, including the pedestrians and the bus in the background. Our estimated kernel is also the most accurate among all the methods.

et al., 2008; Krishnan et al., 2011] and the variational Bayesian ones [Fergus et al., 2006; Levin et al., 2011a], to explicitly recover sharp edges for kernel computation.

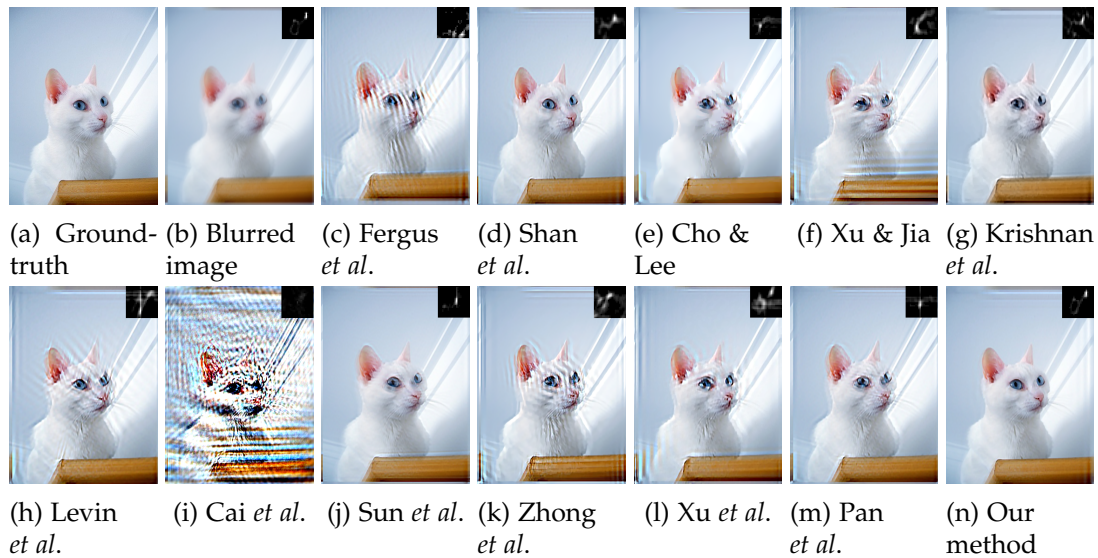


Figure 3.14: Comparisons on a sample image from the Cat dataset[Zhang et al., 2008]. Our method recovers fine texture around the neck, mouth, and whiskers, which cannot be accurately reproduced by the others.

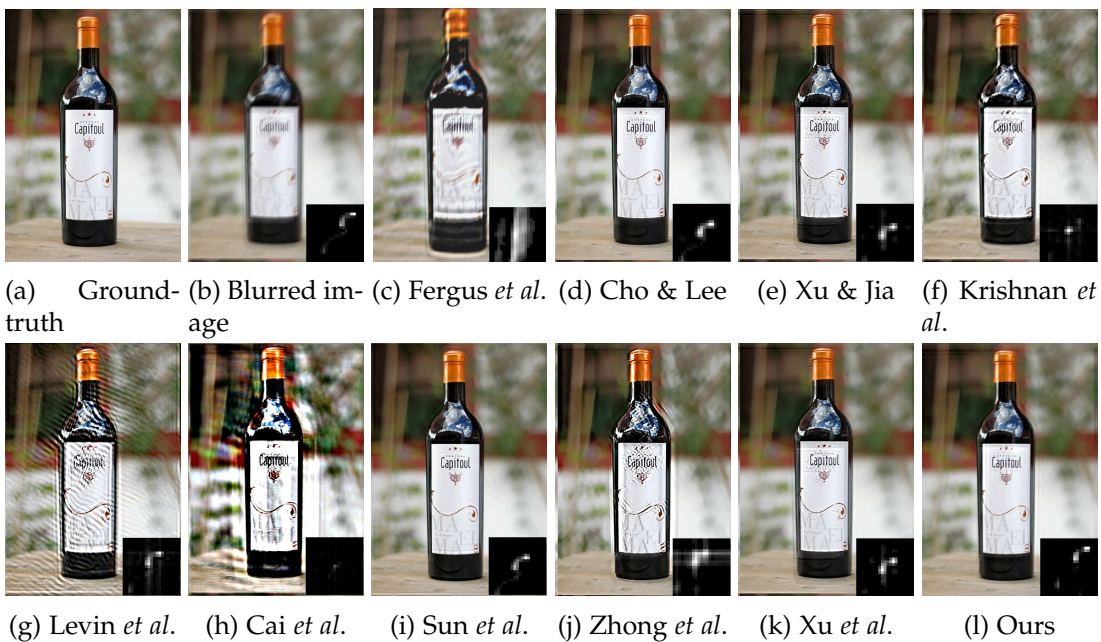


Figure 3.15: Comparisons on a sample image with strong edges and a blurred background, selected from the ETHZ Shape Classes dataset [Ferrari et al., 2010]. The visual quality, *e.g.* sharpness of the text on the label, reproduced by our method is par to the best one among the other methods, *i.e.*[Sun et al., 2013].

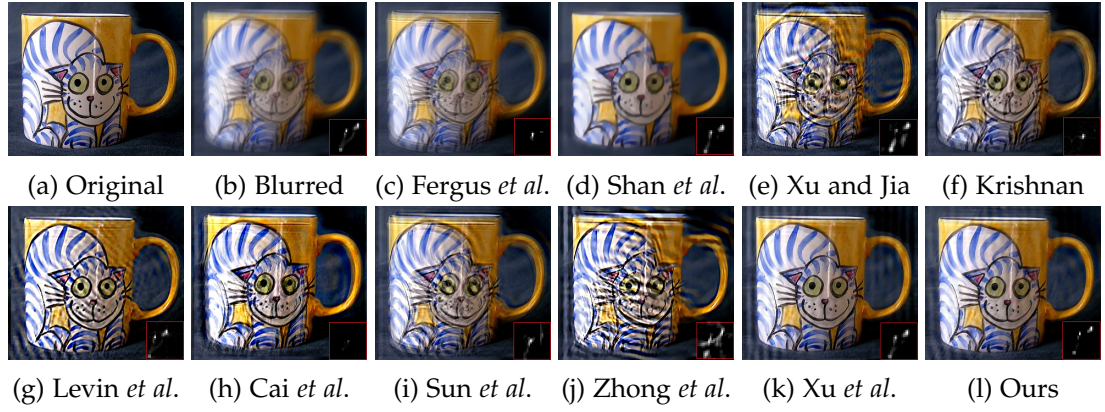


Figure 3.16: Comparisons on a sample image with rich textures, selected from the ETHZ Shape Classes dataset [Ferrari *et al.*, 2010]. On a magnified view, the image our method recovers is sharper than those generated by most of the methods, and comparable to the best, *i.e.* of [Xu *et al.*, 2013], while exhibiting a less degree of ringing artifacts.

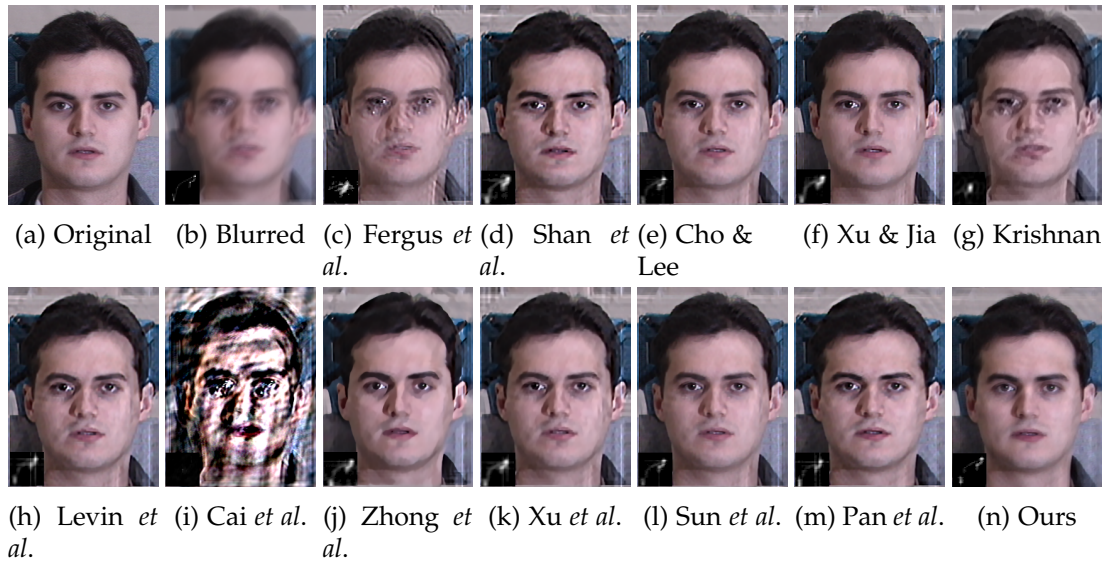


Figure 3.17: Comparisons on a face image selected from the CMU PIE dataset [Sim *et al.*, 2002]. Although our deblurred image appears to be similar to those produced some other methods, its intensity profile (on the face) is richer than the other methods.

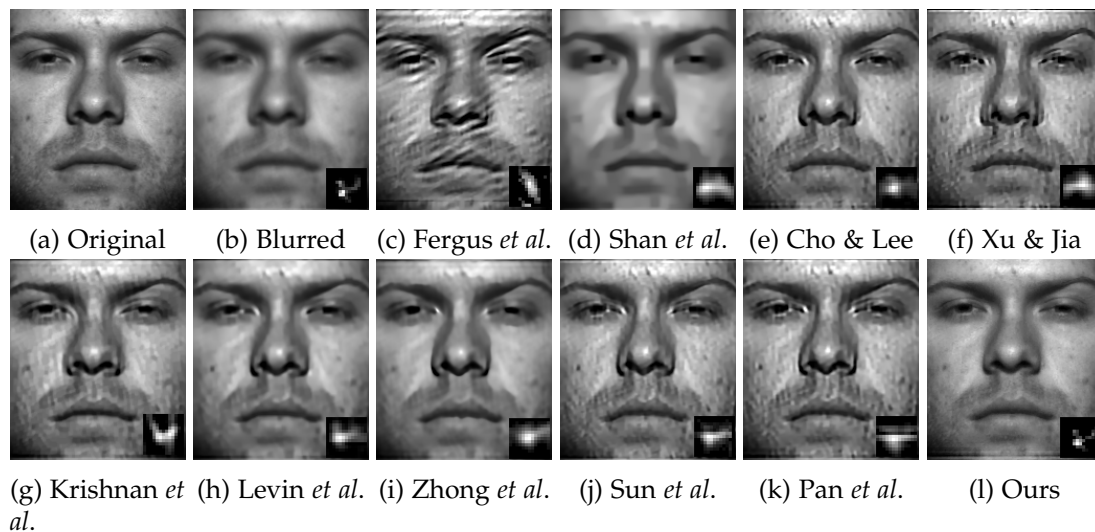


Figure 3.18: Comparisons on a sample face image selected from the Yale-B dataset[Georghiades *et al.*, 2001]. The image we recover is more natural and contains less ringing and exaggerated contrast artifacts. Our estimated kernel is also the closest to the ground-truth.

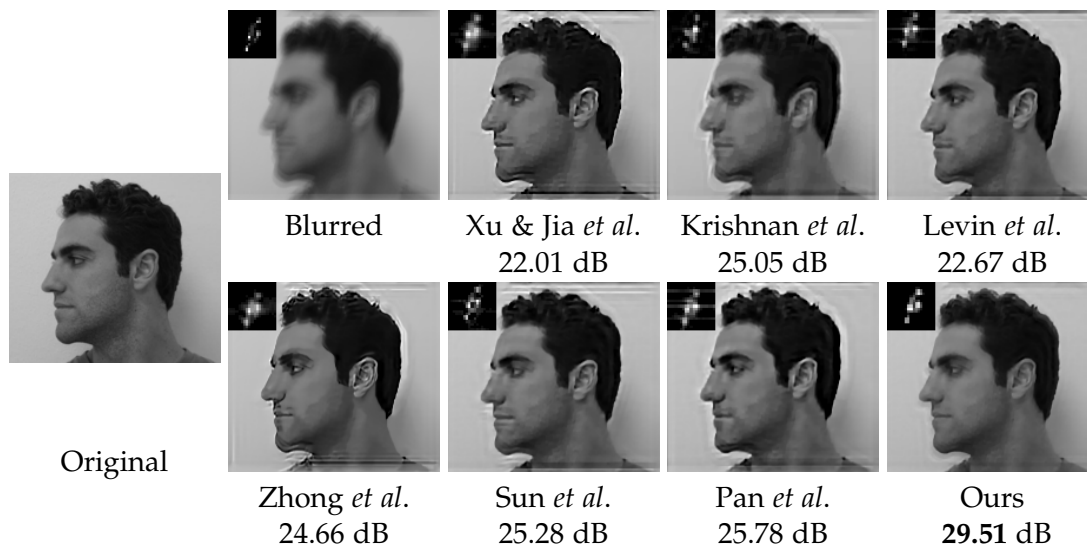


Figure 3.19: Comparisons on a sample image from the FEI dataset[Thomaz and Giraldi, 2010]. Differences can be better seen in magnified view.

In Figure 3.13, we show that our method successfully recovers several objects with a large resemblance to the ground-truth, namely the foreground and background pedestrians, as well as the bus in the background. In contrast, the mentioned objects are unrecognizable in the other deblurred images. Moreover, our estimated kernel is

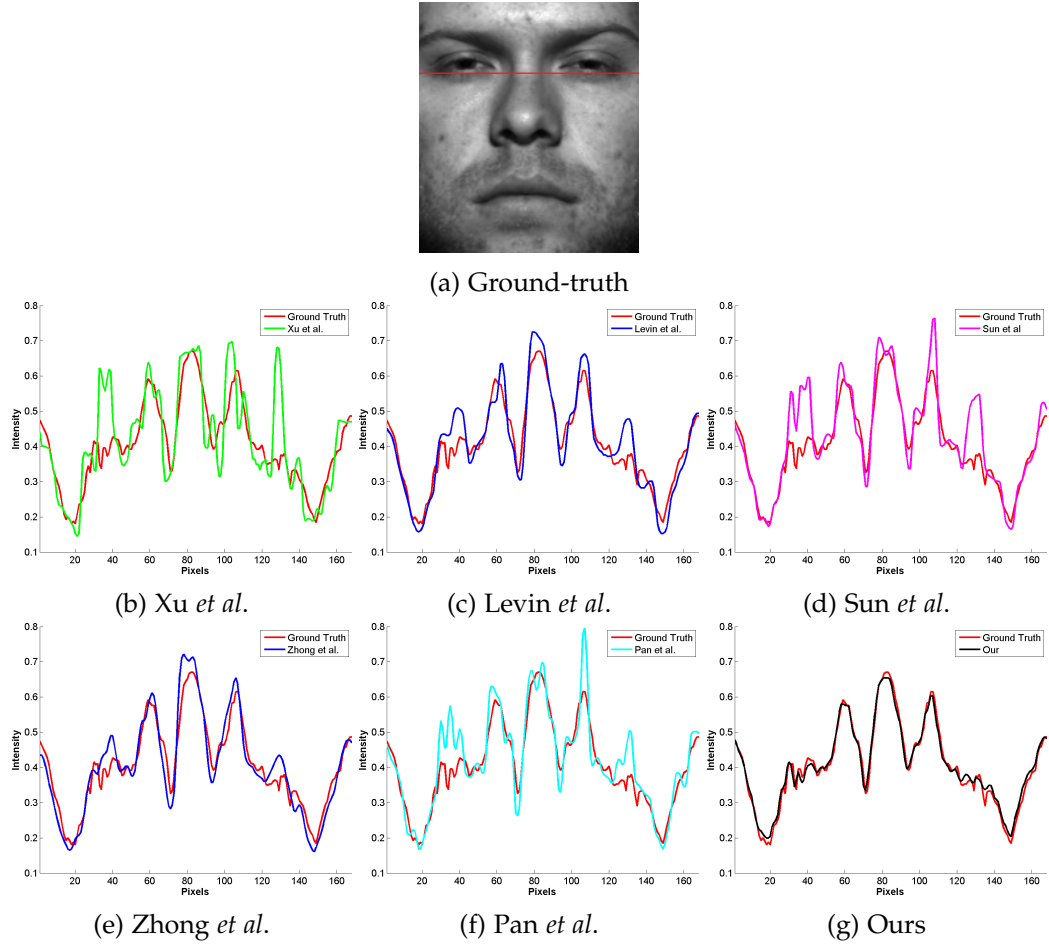


Figure 3.20: Intensity profiles (corresponding to pixel row 55) of the deburred images produced by our method and others. The input blurred image is given in Figure 3.18b. The red trace in each subplot shows the ground-truth profile.

also the most accurate one among all the methods.

Next, we depict an example from the Cat dataset [Zhang *et al.*, 2008]. The visual quality of our recovered image, as shown in Figure 3.14n outperforms all others. Noticeable features restored by our method include the sharpness and the clarity of the cat’s eyes. Our method can also recover subtle textures around the neck, mouth, and whiskers of the cat while they are not reconstructed in the results produced by the other methods. Further, a magnified view of the results in Figures 3.14e-3.14m shows that the methods that rely on edges, *e.g.* [Krishnan *et al.*, 2011], and patches with high-contrast, *e.g.* [Sun *et al.*, 2013], fail to yield an accurate estimation of the kernel. Our method outperforms [Pan *et al.*, 2014a]’s, which uses class exemplars with additional manually drawn mask input around the contours of the cat’s head, the mouth, and eyes in the ground-truth image.

Subsequently, we examine the visual quality of the results achieved by our method and several others on two examples from the ETHZ Shape Classes dataset [Ferrari et al., 2010]. Firstly, in Figure 3.15, we show results for an image containing strong occlusion and text edges and a blurred background with no sharp textures. Again, our method can recover reasonably sharp edges and text in the image. Meanwhile, the methods of [Fergus et al., 2006], [Xu and Jia, 2010], [Krishnan et al., 2011], [Levin et al., 2011a], [Cai et al., 2012], [Zhong et al., 2013] and [Xu et al., 2013] have poorly estimated the PSF, which indirectly causes ringing artifacts and multiple false edges in the deblurred image. At close inspection, [Cho and Lee, 2009] method produces slight ringing on the left occlusion boundary of the bottle and false edges on the white background of the label. Meanwhile, Sun *et al.*'s result in Figure 3.15i appears to be comparable to ours, although the kernel they recover incurs a downward shift compared to the ground-truth. Secondly, we qualitatively compare deblurring results for an image with rich textures and edge information, as shown in Figure 3.16. The blurred edges in the input image 3.16b are of different thicknesses, which potentially causes incorrect estimation of the kernel. Therefore, methods relying on strong or thick edges such as [Fergus et al., 2006], [Xu and Jia, 2010], [Krishnan et al., 2011], [Levin et al., 2011a] result in strong ringing artifacts and incorrect edges. [Shan et al., 2008]'s method suffers less ringing artifacts than the others, but the result appears to be over-smoothed. On a magnified view, the image we recover is sharper than those produced by most of the other methods, and comparable to the best of them, *i.e.* of [Xu et al., 2013], while exhibiting a lower level of ringing artifacts.

Further, we examine the results for two benchmark face image datasets. The first one is taken from the CMU PIE face dataset [Sim et al., 2002]. As shown in Figure 3.17b, the blurred image is quite challenging due to the large scale and the complex trajectory of the blur-induced motion. As a result, [Fergus et al., 2006], [Shan et al., 2008], [Krishnan et al., 2011], [Levin et al., 2011a], [Xu et al., 2013] and [Pan et al., 2014a] yields incorrect kernels which are less sparse than the ground-truth one. Although our deblurred image appears to be similar to those of [Xu and Jia, 2010], [Zhong et al., 2013], [Xu et al., 2013] and [Sun et al., 2013], it shows gradual changes in the intensity across the face, as opposed to the flatness on the other deblurred images. This result suggests that our algorithm has recovered a wider range of spatial frequencies than the high frequencies reproduced by the other methods.

Another face example is selected from the Yale-B dataset [Georghiades et al., 2001], which contains cropped and well-aligned face images. In Figure 3.18, note that Xu *et al.*'s result is not available for this example since the image dimensions of less than 200×200 pixels are below the limit that can be handled by their implementation. The methods of [Fergus et al., 2006; Cho and Lee, 2009; Xu and Jia, 2010; Krishnan

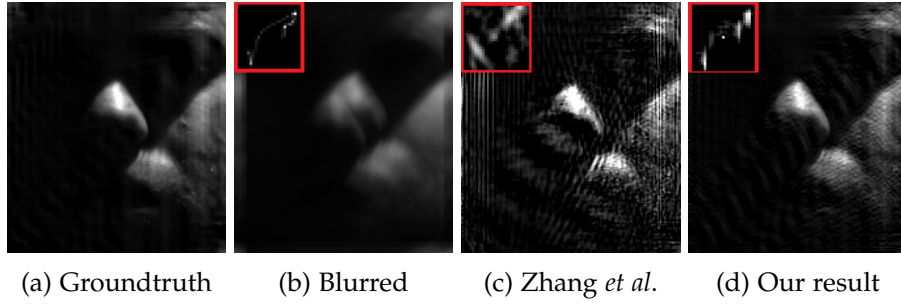


Figure 3.21: Comparison with [Zhang et al., 2011]. (a) Ground-truth image, (b) blurred image, (c) deblurring results produced by [Zhang et al., 2011], which is reported as a failure case in their paper, (d) our results.

et al., 2011; Sun et al., 2013; Pan et al., 2014a] emphasize noise and artifacts and estimate the kernel incorrectly. Meanwhile, the images estimated by [Shan et al., 2008], [Levin et al., 2011a] and [Zhong et al., 2013] are either over-smooth or lack fine details such as hair and speckles.

To visually demonstrate that our method recovers the image more accurately, we randomly select a row of pixels from the ground-truth image and compare it with the corresponding row in the recovered image. In Figure 3.20, it can be observed that our method is the closest to the ground-truth scanline. The failure of the alternative methods is partly due to the lack of strong edges in the blurred input image. On the other hand, by taking the learned frequency spectrum of faces, our method can recover more curvature and finer details in the face, and less ringing artifacts than the others. Our estimated kernel is also the closest to the ground-truth compared to the others.

Furthermore, our method is not restricted to only frontal face images, but can also deblur face images in a different view. In Figure 3.19, we show results for an image from the FEI dataset [Thomaz and Giraldi, 2010]. The training data contains images with frontal as well as different viewing angles. Our method yields the highest accuracy (PSNR) using a training dataset comprising images in a similar pose.

3.4.4 Comparison with Exemplar-based Methods

For completeness, we compare our algorithm to several state-of-the-art deblurring methods that use class exemplars. Since the implementations of these methods are not available publicly, the comparisons are purely based on the qualitative results reported in the respective papers. In the comparison, we consider the methods of [Zhang et al., 2011], [Hacohen et al., 2013], and those of [Pan et al., 2014b,a].

As shown in Figure 3.21a), we examine the performance of our method and com-

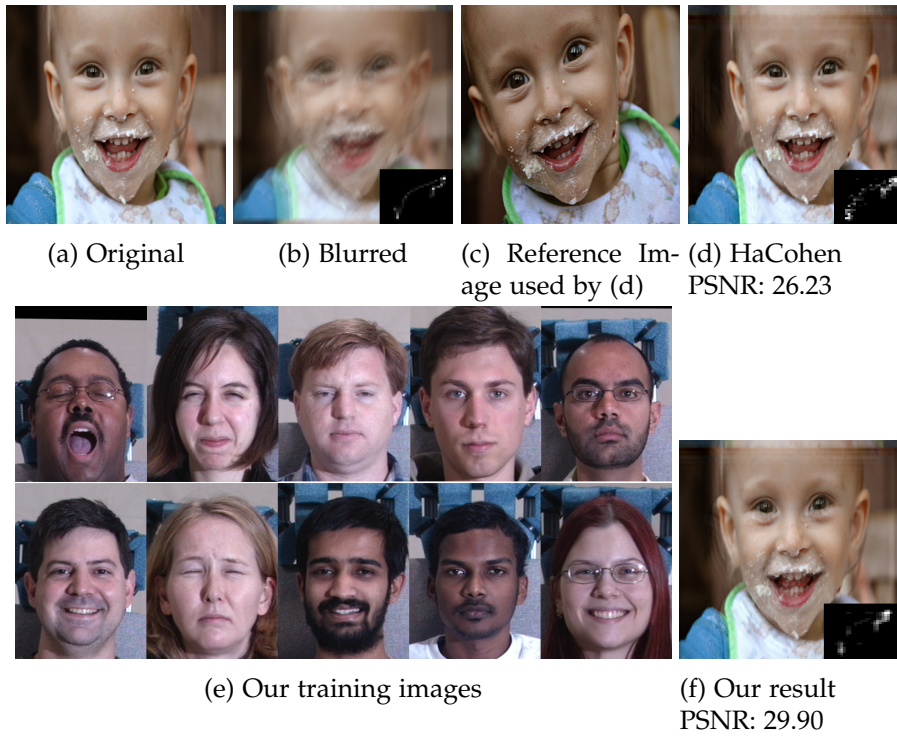


Figure 3.22: Comparison to [Hacohen et al., 2013]’s on a blurred image taken from their paper.

pare it directly with that of [Zhang et al., 2011] from their paper, which they classified as a failure case. This is a challenging example as most of the pixels are dark and noise prone, and there are almost no salient edge features to estimate the kernel correctly. In Figure 3.21b, we show a blurred image generated by the ground-truth kernel depicted at its top left corner. In addition, we obtain the deblurring result recovered by [Zhang et al., 2011]’s algorithm directly from their paper and display both the latent image and the kernel in Figure 3.21c (top-left corner). Similarly, our deblurring result is shown in Figure 3.21d. As visible, the latent image produced by [Zhang et al., 2011] suffers from ringing artifacts, and the estimated kernel does not resemble the sparse structure of the ground-truth. In contrast, our recovered latent image contains much fewer artifacts and is visually closer to the ground-truth. Furthermore, the kernel computed by our algorithm appears to be more similar in shape to the original kernel and much sparser than that produced by [Zhang et al., 2011].

Next, we illustrate the advantage of our method over that by [Hacohen et al., 2013] by an example taken directly from their paper. This method requires a dense correspondence to be established between the blurred input image and a reference image of the same content and structure. Here, it is observed that our result is



Figure 3.23: Deblurring of an image containing foreground text and complex background. (a) Ground-truth (sharp) image, (b) blurred image, (c) deblurring results by [Pan et al., 2014b], (d) our results (zoom in to see the differences).

comparable to the other method. However, the greatest gain from our method is the simplicity of the required input. Our method does not need a reference image with restrictive content and structure and a correspondence map to the blurred image. The only requirement for our input is that the training images belong to the same class. In this example, [Hacohen et al., 2013] employed a reference image of the same person, with many matches to the blurred image, whereas our method permits the flexibility to collect training faces images of various individuals and expressions.

More recently, [Pan et al., 2014b] introduced a deblurring algorithm for two-tone text images using an ℓ_0 -regularized intensity and gradient prior and applied it to the deblurring of non-document text images. We examine the performance of their method on an image from ETHZ shape classes dataset [Ferrari et al., 2010]. As shown in Figure 3.23a, the image contains a cup with large printed text in the foreground and some cluttered background regions, which possess the same features as the non-document text images used in their paper. Comparing our deblurring result in Figure 3.23d to that of [Pan et al., 2014b] in 3.23c, we observe that our estimated kernel is more similar to the ground-truth kernel than the other. Moreover, in the image recovered by Pan *et al.*, ringing artifacts are visible around the edges of the text printed on the cup. In contrast, our algorithm recovers the foreground text with increased sharpness and much less ringing than the former method. Furthermore, it restores the legibility of the background text within the marked red box, which [Pan et al., 2014b] failed. This stems from the fact that, the ℓ_0 -regularized prior employed by Pan *et al.* simply favors uniform intensity regions, which is insufficient to capture spatial variations caused by illumination and shading in shape images. In this example, our method has aimed to capture this variation via the subspace of frequency bands and therefore is more successful in restoring the original image.

Subsequently, we compare our method with [Pan et al., 2014a], which aims at face image deblurring using annotated salient edges of sharp exemplars. For a fair

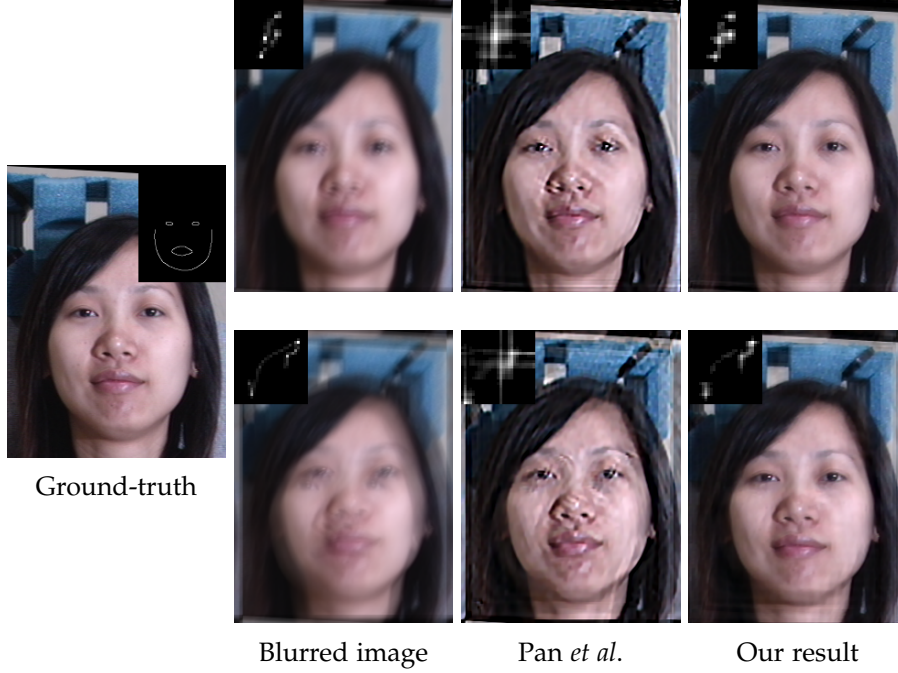


Figure 3.24: Results for an image provided by [Pan et al., 2014a]. First column: ground-truth image, second column: blurred images and original blur kernels (at the top left corners of the images), third column: deblurred images and estimated kernels by [Pan et al., 2014a], fourth column: our results.

comparison, we use the dataset and the mask annotations provided by the authors. In Figure 3.24, we depict the deblurring results delivered by both methods for an example ground-truth image (the first column), which are blurred with two different blurred kernels (the second column). Here, we run their implementation with the contour mask obtained from the ground-truth image as shown at the top left corner in the first column of Figure 3.24. This mask, according to [Pan et al., 2014a], plays a major role in the selection of salient edges as input for their method. The recovered images by [Pan et al., 2014a] and our method are shown in the third and fourth columns, respectively. We also show the estimated kernel at the top-left corners of these images. Comparing these results, our method produces a sharper image with much fewer artifacts. Besides, it can be seen that our kernels exhibit a strong similarity to the ground-truth ones.

3.5 Discussion

We have introduced a novel class-specific prior that significantly improves the performance of image deblurring. The prior is designed to capture the properties of trans-

form domain coefficients for specific image classes over the entire spectrum of frequency bands. Representing images on the class-specific subspaces, we reconstruct the frequency responses suppressed after the blurring process. Our approach overcomes the limitation of existing methods when dealing with blurred images lacking high-frequency details. In this chapter, we have incorporated the external datasets in the formulation and presented a closed form solution for image deblurring.

We provided an insight into our algorithm and discussed each component and its effect on deblurring results in detail. We have also shown the convergence of our method and the running time is taken for deblurring a typical image. Furthermore, we have demonstrated the role of this prior in extensive experimental evaluations. We show that our method outperforms prior deconvolution works that use generic priors and class exemplars both in numerical accuracy and visual quality. We have provided ample visual examples in this chapter to show the superior performance of our method.

Although our algorithm performs well; however, there are scenarios in which our approach underperforms due to several factors. For example, high noise in the blurry image, a considerable blur kernel and nonlinear camera response function which is unknown. Similarly, our approach may not produce desirable results when the blurry image and training images do not correspond.

Category-Specific Object Image Denoising

I never think of photographs as being individual. Always as a group.

Martin Parr

We present a novel image denoising algorithm that uses external, category specific image database. In contrast to existing noisy image restoration algorithms that search patches either from a generic database or noisy image itself, our method first selects clean images similar to the noisy image from a database that consists of images of the same class. Then, within the spatial locality of each noisy patch, it assembles a set of “support patches” from the selected images. These noisy-free support samples resemble the noisy patch and correspond principally to the identical part of the depicted object. In addition, we employ a content adaptive distribution model for each patch where we derive the parameters of the distribution from the support patches. We formulate noise removal task as an optimization problem in the transform domain. Our objective function composed of a Gaussian fidelity term that imposes category specific information, and a low-rank term that encourages the similarity between the noisy and the support patches in a robust manner. The denoising process is driven by an iterative selection of support patches and optimization of the objective function. Our extensive experiments on five different object categories confirm the benefit of incorporating category-specific information to noise removal and demonstrates the superior performance of our method over the state-of-the-art alternatives.

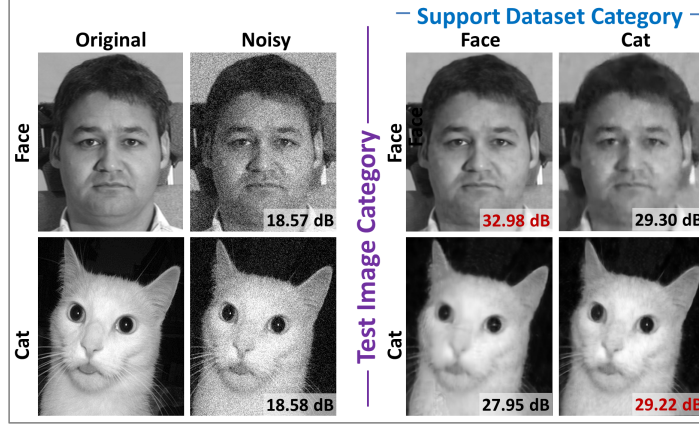


Figure 4.1: Denoising results of two sample images from *face* and *cat* categories. As visible, by using the same category support dataset we generate higher PSNR scores—shown in red (best viewed in high-resolution).

4.1 Introduction

Our objective in this work is to remove noise from images (or image regions) that depict a single object of a known class. This goal perfectly complements the recent advancements in object detection and classification [Girshick et al., 2014; Krizhevsky et al., 2012]. Denoising of images with known classes is instrumental in various applications such as face image enhancement thus all image solutions tasks where face images are used, document image recovery, digital heritage, cell image analysis, and image aesthetics to count a few.

In this chapter, we propose a denoising method for images of single objects using a noise-free external image dataset of the same category. Unlike the existing methods that ignore the relative locations of external patches with respect to the whole object window, our method considers the object part semantics during patch selection by limiting the search to a part-based locality in the most relevant external images, aiming to make the best use of the part-whole relationships.

Our formulation is unique and differs from previous approaches such as [Luo et al., 2015]. Our decision to express the denoising problem in the transform domain is strongly motivated by the need to establish a patch similarity metric that is invariant to the local pixel intensity. In natural images, it is rare that two patches have identical intensity values, but it is common that patches share similar local features such as uniform areas, smooth gradients, edges, corners, and textures. Such local patch features are closely related to the gradient responses and hence can be better represented by frequency coefficients in the transform domain.

We achieve robustness to object pose and scale variations by operating on the patch level, a similar analogy to the part-based models of object classification, and by creating copies of the dataset at various object scales and determining the correct scale for the input image, which can be provided by an object detector.

Figure 4.1 illustrates the imperative role of category-specific datasets in denoising. As visible, a significant improvement in image sharpness and PSNR is achieved when using the correct class dataset for denoising images of known classes. The novel contributions of our approach are as follows.

- A strategy for finding similar external patches to a given noisy patch within the same object part, which we term “support patches” hereafter.
- A formulation of the object category-specific patch denoising problem in a transform domain.
- A Gaussian model of the membership likelihood to a support patch group for a noisy patch.
- A low-rank constraint to enforce the similarity between the noisy patch and its support patches.

This chapter is organized as follows. In Section 4.2, we formulate the image denoising problem by incorporating new class-specific priors. Section 4.3 presents an optimization approach to the denoising problem. Furthermore, in Section 4.4, we discuss the components of our method and then present a detailed comparison with current state-of-the-art methods. Finally, we conclude our class-specific denoising work in 4.5.

4.2 Denoising Problem Formulation

The noisy image model relates the true pixel value $\text{vec}(\mathbf{x})$ to the noisy value $\text{vec}(\mathbf{y})$ at the same pixel by

$$\text{vec}(\mathbf{y}) = \text{vec}(\mathbf{x}) + \eta, \quad (4.1)$$

where $\eta \sim \mathcal{N}(0, \sigma_n^2)$ is assumed to be Gaussian noise with a standard deviation σ_n .

We consider the problem of recovering the latent (true) image, given the noisy image of an object and a dataset of noise-free images in the same object category. Let the matrices $\mathbf{X} \in \mathbf{R}^{n \times m}$, $\mathbf{Y} \in \mathbf{R}^{n \times m}$ represent the pixel values of the true and observed images, and the set of matrices $\{\mathbf{Z}_k : k = 1, \dots, K\}$ denote the external dataset.

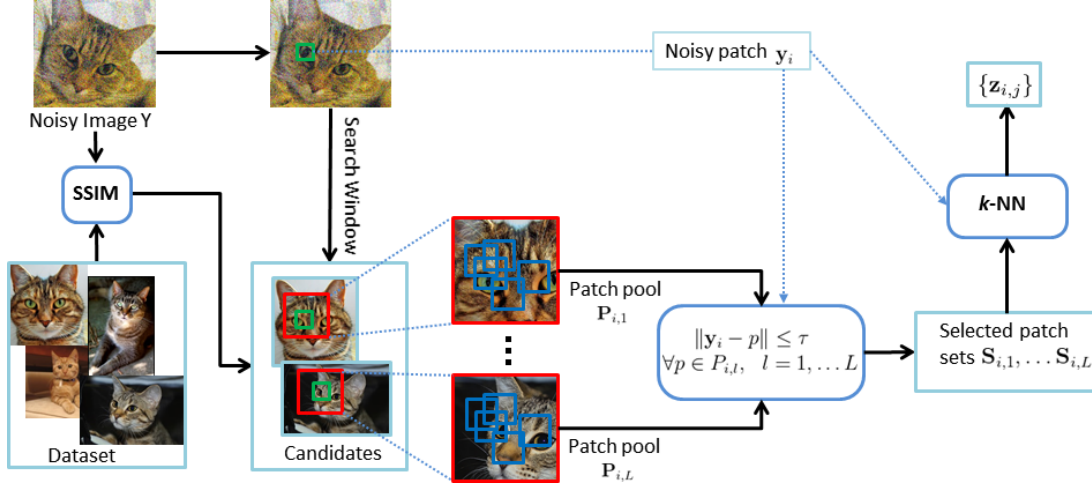


Figure 4.2: Searching and selecting support patches for a given noisy patch y_i . Candidate images similar to the noisy image (measured by SSIM) are selected from the given database. Subsequently, in each candidate image, we search for patches that are similar to the noisy patch, *i.e.* within a Euclidean distance of τ from y_i . The search is restricted to a local window in each candidate image. Finally, among the remaining patches, only the nearest neighbors to y_i are retained for denoising.

4.2.1 Support patch search

As mentioned earlier, image denoising by collaborative filtering relies on the similarity between patches and performs aggregation of their denoised version. Following this approach, we collect all the overlapping patches of the noisy image, and denote the intensity vector of the patch centered at the i -th pixel by y_i $i = 1, \dots, M$. Likewise, x_i denotes the patch intensity vector for the corresponding location in the latent image X .

For a given noisy patch, we find the most similar “support patches” from the dataset of category-specific images. Due to the similarity in their local features, support patches facilitate noise suppression by enforcing a group sparsity constraint. Various approaches such as BM3D [Dabov et al., 2007b] and non-local means (NLM) denoising [Buades et al., 2005; Goossens et al., 2008; Lebrun et al., 2013] have employed local and non-local patch similarity to separate the latent structure of a patch from its noise component.

In our algorithm, the support patch selection occurs in several stages. Firstly, we select a preset number (L) of external images that are structurally most similar to the noisy image based on the structural similarity (SSIM) index. Subsequently, from the l -th candidate image ($l = 1, \dots, L$), we obtain a pool $P_{i,l}$ of patches that are similar to a given noisy patch y_i .

We take into account the difference in resolution and aspect ratio between the input and the candidate image when determining the local search window. Suppose that H and W are the height and width of the input image, and H_l and W_l are the corresponding quantities of the candidate image. The center coordinates $[r_{i,l}, c_{i,l}]^T$ of the local search window in the l -th candidate image is a linear mapping of the location $[r_i, c_i]^T$ of the i -th pixel as follows, $r_{i,l} = \lfloor r_i \frac{H_l}{H} \rfloor$ and $c_{i,l} = \lfloor c_i \frac{W_l}{W} \rfloor$. The search window has a preset size of 51×51 .

Finally, within each patch pool $P_{i,l}$ we only retain those that have a Euclidean distance from the input patch y_i that is below a threshold τ . We denote the resulting set of refined patches by $S_{i,l}$. Next, we aggregate the refined patch pools $S_{i,l}$ across the candidate images. Within the resulting collection, we perform a k -NN search for the most similar patches to y_i . In the end, we obtain a set of support patches $\{z_{i,j} : j = 1, \dots, T_i\}$ resembling the noisy patch y_i .

Figure 4.2 shows the procedure for searching and selecting support patches for a given noisy patch. In the figure, the noisy patch y_i is bounded by a green rectangle (top row) while the patches with blue boundaries illustrate members of the patch pools $P_{i,l}$. In both the noisy and candidate images, the search space is indicated by a red rectangular boundary.

4.2.2 Transform domain formulation

In our formulation, we opt to represent local patches in a transform domain, rather than the patch intensity domain. This is because matching patches in the original space of patch intensity vectors are susceptible to a bias in the overall patch intensity, such as local illumination. Representing patches in the transform domain encourages matching between those that have a various range of intensity but similar local structure.

To improve robustness to patch intensity, we subtract the mean patch intensity from the patch intensity vector before performing the domain transform. The per-patch mean subtraction effectively removes the zero-frequency (DC) bias, yielding patches lying in a $N - 1$ -dimensional subspace, where N is the number of patch pixel. Therefore, the latent patch can be represented by the remaining $D = N - 1$ transform coefficients. The Gaussian prior and the low-rank constraint in the space of non-zero-frequency coefficients effectively enforce patch similarity and are less susceptible to variations in patch intensity.

With this intention, we introduce the notation for representing images and patches in the discrete transform domain. Here, we choose to use the DCT transform, although other popular transforms such as the wavelet, Fourier, DST and the Walsh-Hadamard transforms can be employed for the same purpose.

We denote the patch transform by $\mathcal{T} : \mathbb{R}^N \rightarrow \mathbb{R}^D$ that maps the original N -dimensional intensity vector to a vector of $D = N - 1$ non-zero frequency DCT coefficients, with N being the number of pixels per patch. Let $\Phi \in \mathbb{R}^{N \times D}$ denote the DCT basis spanning this D -dimensional subspace. Note that $\Phi^T \Phi = \mathbf{I}$. Let us also denote the mean subtracted versions of the patches \mathbf{x}_i and \mathbf{y}_i by $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{y}}_i$, respectively. The intensity vector of these patches are related to the transform coefficient vectors α_i and $\beta_i \in \mathbb{R}^D$ by $\alpha_i = \Phi^T \bar{\mathbf{x}}_i$, $\beta_i = \Phi^T \bar{\mathbf{y}}_i$, and $\bar{\mathbf{x}}_i = \Phi \alpha_i$, $\bar{\mathbf{y}}_i = \Phi \beta_i$. Once the transformation coefficients α_i is estimated, we can compute the mean-subtracted latent patch by $\bar{\mathbf{x}}_i$ by an inverse DCT transform and recover the original patch \mathbf{x}_i by adding its intensity mean to $\bar{\mathbf{x}}_i$.

Next, we will formulate the various components of our denoising problem.

4.2.3 Data Fidelity

Assuming the independence of individual pixel values, the conditional likelihood of the noisy image given the original (noise-free) image is

$$p(\mathbf{Y}|\mathbf{X}) \propto \exp \left(-\frac{\|\mathbf{Y} - \mathbf{X}\|_2^2}{\sigma_n^2} \right), \quad (4.2)$$

where $\|\cdot\|_2$ stands for the ℓ_2 norm of a vector.

The reconstruction of the noise-free image \mathbf{X} aims to maximize the conditional log likelihood in Equation 4.2, which is equivalent to minimizing the data fidelity term $\|\mathbf{Y} - \mathbf{X}\|_2^2$, which is a sum of squared errors over image pixels. Since each image pixel belongs an approximately equal number of overlapping patches, *i.e.* N , the term above can be approximated as a multiple of the sum of these terms evaluated on a per-patch basis, *i.e.* $\|\mathbf{Y} - \mathbf{X}\|_2^2 \approx \frac{1}{N} \sum_{i=1}^M \|\mathbf{y}_i - \mathbf{x}_i\|_2^2$. Furthermore, $\mathbf{y}_i - \mathbf{x}_i = \bar{\mathbf{y}}_i - \bar{\mathbf{x}}_i$ assuming that the mean intensity of the latent patch \mathbf{x}_i is estimated from that of \mathbf{y}_i , and $\|\bar{\mathbf{y}}_i - \bar{\mathbf{x}}_i\|_2^2 = \|\Phi(\alpha_i - \beta_i)\|_2^2 = \|\alpha_i - \beta_i\|_2^2$ due to the orthonormality of the basis Φ . Expressing the data fidelity in terms of the transform coefficients, we obtain

$$\|\mathbf{Y} - \mathbf{X}\|_2^2 \approx \frac{1}{N} \sum_{i=1}^M \|\alpha_i - \beta_i\|_2^2. \quad (4.3)$$

4.2.4 Support Patch Group Membership

Now we define an additional constraint that imposes the similarity between a noisy patch and those from an image dataset belonging to the same object category. Recall that the patch search described in Section 4.2.1 results in T_i support patches $\{\mathbf{z}_{i,j} : j = 1, \dots, T_i\}$ that resemble the local appearances of the noisy patch \mathbf{y}_i . Here, we rely on the statistics of the support patch group $\mathbf{z}_{i,j}$ in the transform domain in

order to predict the latent patch \mathbf{x}_i from \mathbf{y}_i . Let the transform coefficients of $\mathbf{z}_{i,j}$ be $\{\gamma_{i,j} : j = 1, \dots, T_i\}$, where T_i is the number of support (most similar) patches of patch \mathbf{x}_i , and μ_i and Σ_i be the mean and covariance matrix estimated from these transform coefficient vectors.

Assuming that similar patches belong to a Gaussian distribution in the transform domain, the most probable \mathbf{x}_i is one that maximizes its likelihood of belonging to the support patch group, *i.e.* $p(\alpha_i | \mu_i, \Sigma_i) \propto \exp\left(-\frac{1}{2}(\alpha_i - \mu_i)^T \Sigma_i^{-1} (\alpha_i - \mu_i)\right)$. This is equivalent to minimizing the log-likelihood

$$\log p(\alpha_i | \mu_i, \Sigma_i) \propto \frac{1}{2}(\alpha_i - \mu_i)^T \Sigma_i^{-1} (\alpha_i - \mu_i), \quad (4.4)$$

which is the Mahalanobis distance from a noisy patch to the distribution of its support patches in the transform domain.

4.2.5 Low-rank Constraint

We further formulate a low-rank constraint concerning a noisy patch and its support patches. The intuition behind this constraint is that the local structure of a patch can be sparsely represented by a basis with low cardinality. Therefore, when similar patch vectors are stacked as columns of a matrix, the matrix should exhibit the low-rank property and have sparse singular values. In [Dong et al., 2013], the authors, derived this low-rank property directly from the common observation that the structural similarity between patches can be encoded as a group sparsity constraint, in terms of the $\ell_{p,q}$ norm of the above matrix.

However, the rank minimization problem is NP-hard and thus is intractable to solve directly. In their work, [Candès and Recht, 2009] have provably derived the tightest convex relaxation of the rank minimization problem in the form of a matrix nuclear norm minimization problem. Under certain conditions, these two problems have the same unique solution. Therefore, the low-rank approximating matrix can be recovered exactly by solving the nuclear norm minimization (NNM) problem.

To formulate the NNM problem, for each latent patch \mathbf{x}_i , we form a data matrix \mathbf{M}_i containing its transform coefficients and those of its support patches as its columns, as $\mathbf{M}_i = [\alpha_i, \gamma_{i,1}, \dots, \gamma_{i,T_i}]$. Here, we aim to minimise the matrix nuclear norm $\|\mathbf{M}_i\|_*$, which is the sum of its singular values.

4.3 Optimization

In previous the sections, we have described the data fidelity term in Equation 4.3, the patch group membership term in Equation 4.4 and the nuclear norm constraint on

\mathbf{M}_i for each noisy patch \mathbf{y}_i . Aggregating all these terms over all the image patches $\mathbf{y}_i, i = 1, \dots, M$, we formulate the overall minimization problem as

$$\mathcal{L} = \sum_{i=1}^M \mathcal{L}_i, \quad (4.5)$$

where the term \mathcal{L}_i is related to only the i -th noisy patch as

$$\mathcal{L}_i = \frac{1}{\sigma_n^2} \|\alpha_i - \beta_i\|_2^2 + \lambda_1 (\alpha_i - \mu_i)^T \Sigma_i^{-1} (\alpha_i - \mu_i) + \lambda_2 \|\alpha_i, \gamma_{i,1}, \dots, \gamma_{i,T_i}\|_*, \quad (4.6)$$

where $\{\gamma_{i,j} : j = 1, \dots, T_i\}$ are the transform coefficients of the support patches for the patch \mathbf{x}_i , $\|\cdot\|_*$ is the nuclear norm of a matrix and λ_1 and λ_2 are the weights of the support patch group likelihood and the nuclear norm terms.

4.3.1 Patch Denoising

We can minimize the overall objective function in Equation 4.5 by minimizing each of the term \mathcal{L}_i independently. To this end, we introduce an auxiliary variable $\mathbf{M}_i \triangleq [\alpha_i, \gamma_{i,1}, \dots, \gamma_{i,T_i}]$ to Equation 4.6. Subsequently, we relax the equality constraint as minimising the squared Frobenius norm $\|\mathbf{M}_i - [\alpha_i, \gamma_{i,1}, \dots, \gamma_{i,T_i}]\|_F^2$ and incorporate it into the objective function.

In addition, we normalize the term by a Lagrange multiplier equal to $\frac{1}{(T_i+1)\sigma_n^2}$, which accounts for the image noise and the number of support patches. For a patch \mathbf{x}_i , we then minimize \mathcal{L}_i with respect to the transform α_i and the variable \mathbf{M}_i

$$\begin{aligned} (\alpha_i^*, \mathbf{M}_i^*) = \operatorname{argmin}_{\alpha_i, \mathbf{M}_i} & \frac{1}{\sigma_n^2} \|\alpha_i - \beta_i\|_2^2 + \lambda_1 (\alpha_i - \mu_i)^T \Sigma_i^{-1} (\alpha_i - \mu_i) \\ & + \frac{\|\mathbf{M}_i - [\alpha_i, \gamma_{i,1}, \dots, \gamma_{i,T_i}]\|_F^2}{(T_i + 1)\sigma_n^2} + \lambda_2 \|\mathbf{M}_i\|_*. \end{aligned} \quad (4.7)$$

The relaxed objective function in Equation 4.7 is convex with respect to α_i and \mathbf{M}_i separately, while the other variable is fixed. More specifically, when \mathbf{M}_i is fixed, the non-constant terms, including the squared Frobenius norm, are quadratic functions of α_i . On the other hand, when α_i is fixed, the objective function involves a nuclear norm of \mathbf{M}_i and a squared Frobenius norm. It is known that the nuclear norm is convex in the space of the matrix \mathbf{M}_i , and the squared Frobenius norm is regarded as a quadratic function of the matrix elements.

We employ an iterative procedure to minimise the cost function in Equation 4.7. Each iteration involves an alternating optimization scheme concerning either α_i or \mathbf{M}_i , while fixing the other. Since each of these steps aims to solve a convex sub-

problem with respect to its variable, this scheme is guaranteed to converge to a global minimum in each step, with respect to either α_i or \mathbf{M}_i .

4.3.1.1 Update of α_i with Fixed \mathbf{M}_i

With a fixed value of \mathbf{M}_i^* at the current iteration, we solve the sub-problem

$$\alpha_i^* = \underset{\alpha_i}{\operatorname{argmin}} \frac{\|\alpha_i - \beta_i\|_2^2}{\sigma_n^2} + \lambda_1(\alpha_i - \mu_i)^T \Sigma_i^{-1}(\alpha_i - \mu_i) + \frac{\|\alpha_i - \mathbf{M}_i^*(:,1)\|_2^2}{(T_i + 1)\sigma_n^2}, \quad (4.8)$$

where $\mathbf{M}_i^*(:,1)$ denotes the first column of the matrix \mathbf{M}_i^* . Since the problem above is quadratic in α_i , taking its derivative leads to the following linear equation, which can be solved by standard techniques.

$$\left(\frac{T_i + 2}{T_i + 1} \mathbf{I} + \lambda_1 \sigma_n^2 \Sigma_i^{-1} \right) \alpha_i^* = \beta_i + \lambda_1 \sigma_n^2 \Sigma_i^{-1} \mu_i + \frac{\mathbf{M}_i^*(:,1)}{T_i + 1}. \quad (4.9)$$

4.3.1.2 Update of \mathbf{M}_i with Fixed α_i

With the values of α_i^* obtained from the previous step, we form a data matrix $\hat{\mathbf{M}}_i \triangleq [\alpha_i^*, \gamma_{i,1}, \dots, \gamma_{i,T_i}]$ for each patch. The sub-problem to be solved with respect to \mathbf{M}_i is then stated as

$$\mathbf{M}_i^* = \underset{\mathbf{M}_i}{\operatorname{argmin}} \|\mathbf{M}_i - \hat{\mathbf{M}}_i\|_F^2 + \tau \|\mathbf{M}_i\|_*, \quad (4.10)$$

where $\tau = \lambda_2(T_i + 1)\sigma_n^2$.

The above problem is related to finding an approximation to a given matrix with a minimal nuclear norm. To solve the problem, we turn our attention to the singular value shrinkage operator developed by [Cai et al., 2010]. Suppose that we have $U\Lambda V^T$ as the singular value decomposition of $\hat{\mathbf{M}}_i$, with Λ_k being the k -th singular value. Theorem 2.1. in [Cai et al., 2010] derives the optimal solution to Equation 4.10 by soft-thresholding the singular values to obtain

$$\mathbf{M}_i^* = U \mathcal{S}_\tau(\Lambda) V^T, \quad (4.11)$$

where the soft-thresholding operator is defined as $\mathcal{S}_\tau(\Lambda) = \operatorname{diag}(\{(\Lambda_k - \tau)_+\})$ with $(x)_+ = \max(x, 0)$.

4.3.2 Recovering Latent Images

Once we have estimated the transform coefficients of individual patches, we recover them in the pixel domain by an inverse transform as $\mathbf{x}_i = \Phi^T \alpha_i, \forall i = 1, \dots, M$ (assum-

ing that Φ is orthonormal). To reconstruct the full image, we translate the patches to their original locations and average the values of overlapping patches at shared pixels. Let R_i denote the patch extraction matrix at the i -th pixel of an image, *i.e.* $\mathbf{x}_i = R_i \mathbf{X}$. With the known matrices R_i 's, the latent image is the optimal solution to the problem

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \lambda_0 \|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{i=1}^M \|R_i \mathbf{X} - \mathbf{x}_i\|_2^2, \quad (4.12)$$

where λ_0 is a positive constant. The least-squares solution to the above equation is

$$\mathbf{X}^* = \left(\lambda_0 \mathbb{I} + \sum_{i=1}^M R_i^T R_i \right)^{-1} \left(\lambda_0 \mathbf{Y} + \sum_{i=1}^M R_i^T \mathbf{x}_i \right). \quad (4.13)$$

The process of patch denoising and latent image recovery occurs iteratively until convergence. Also, we apply the iterative input regularisation technique in [Osher et al., 2005]. Such an approach has been shown to be effective in denoising methods using total variation and wavelets [Xu and Osher, 2007] and spatially adaptive iterative singular value thresholding [Dong et al., 2013]. Specifically, in the t -th iteration, the algorithm takes input from a regularised noisy image $\mathbf{Y}^{(t)}$ computed as follows

$$\mathbf{Y}^{(t+1)} = \mathbf{X}^{(t)} + \rho \left(\mathbf{Y} - \mathbf{X}^{(t)} \right), \quad (4.14)$$

where $\mathbf{X}^{(t)}$ is the current latent image and ρ is a relaxation parameter.

4.3.3 Implementation Details

The overall denoising algorithm consists of interleaving steps of individual patch denoising and whole image restoration. The proposed iterative procedure is summarized in Algorithm 2, with the iteration number denoted by t . As the latent image \mathbf{X} is updated in every iteration, so are the support patches (from the external image dataset) of its patches. Line 5 implements the support patch search procedure described in Section 4.2.1. With the support patches in hand, individual patches in the input image are denoised by alternating the optimization with respect to the variables α_i and \mathbf{M}_i (in lines 10–13). At the end of each iteration, the entire latent image $\mathbf{X}^{(t)}$ is reconstructed from the denoised patches and the input image $\mathbf{Y}^{(t+1)}$ to the next iteration is updated according to Equation 4.14. The noise variance $\varsigma^{(t+1)} \triangleq (\sigma^{(t+1)})^2$ is also updated according to the adjusted input as $\varsigma^{(t+1)} \leftarrow \lambda_n |\varsigma^{(t)} - \frac{1}{M} \|\mathbf{Y} - \mathbf{Y}^{(t+1)}\|^2|$, where M is the number of image pixels and $\lambda_n = 0.17$. The algorithm terminates when the change $\|\mathbf{X}^{(t)} - \mathbf{X}^{(t-1)}\|_2$ falls below a tolerance threshold ϵ .

To improve patch similarity, we follow [Foi et al., 2007] and perform a DCT trans-

Algorithm 2 Denoising with category-specific support patches

Input:

\mathbf{Y} : noisy input image.
 σ_n : noise standard deviation.
 $\lambda_0, \lambda_1, \lambda_2$: term weights in Equations 4.6 and 4.12.
 ρ : relaxation factor in Equations 4.14.
 1: $t \leftarrow 0, \mathbf{X}^{(0)} \leftarrow \mathbf{Y}, \mathbf{Y}^{(0)} \leftarrow \mathbf{Y}, \varsigma^{(0)} \leftarrow \sigma_n^2$.
 2: **repeat**
 3: **for** patch $\mathbf{y}_i^{(t)}$ in $\mathbf{Y}^{(t)}$ **do**
 4: Update $\beta_i^{(t)} \leftarrow \Phi \mathbf{y}_i^{(t)}$.
 5: $\{\mathbf{z}_{i,j}^{(t)} : j = 1, \dots, T_i\} \leftarrow$ support patches of $\mathbf{x}_i^{(t)}$.
 6: **for** $j = 1 \rightarrow T_i$ **do**
 7: Support patch transform $\gamma_{i,j}^{(t)} \leftarrow \Phi \mathbf{z}_{i,j}^{(t)}$.
 8: **end for**
 9: $(\mu_i^{(t)}, \Sigma_i^{(t)}) \leftarrow$ mean and covariance matrix of $\{\gamma_{i,j}^{(t)} : 1 \leq j \leq T_i\}$.
 10: **repeat**
 11: Solve Equation 4.9 for $\alpha_i^{(t)}$.
 12: Update matrix $\mathbf{M}_i^{(t)}$ by Equation 4.11.
 13: **until** Convergence
 14: Update $\mathbf{x}_i^{(t)} \leftarrow \Phi^T \alpha_i^{(t)}$.
 15: **end for**
 16: Reconstruct the image $\mathbf{X}^{(t)}$ by Equation 4.13.
 17: Regularise the input image $\mathbf{Y}^{(t+1)}$ by Equation 4.14.
 18: $t \leftarrow t + 1$.
 19: Update noise variance $\varsigma^{(t+1)} \leftarrow \rho |\varsigma^{(t)} - \frac{1}{M} \|\mathbf{Y} - \mathbf{Y}^{(t+1)}\|^2|$
 20: **until** $\|\mathbf{X}^{(t)} - \mathbf{X}^{(t-1)}\|_2 \leq \epsilon$
 21: **return** Latent (denoised) image $\mathbf{X}^{(t)}$.

form on the mean-subtracted intensity of local patches and subsequently add the mean patch intensities back during patch reconstruction. This effectively means that we only involve the AC components $\gamma_{i,j}$ of the support patches for patch-wise denoising (Section 4.3.1). This technique is based on the observation that subtracting the direct current (DC) component of each patch from its intensity values effectively increases the number of similar local patterns in each group, facilitating a more thorough selection of the most similar support patches to a noisy patch for collaborative filtering.

Further, the per-patch mean subtraction improves the chance of finding a good match, which means a lower number of external images is required for patch-wise denoising.

4.4 Experiments

In this section, we present a detailed performance evaluation of our method against a number of state-of-the-art internal and external image denoising algorithms. Firstly, we examine the influence of the number of category-specific images and support patches on the denoising accuracy. Subsequently, we report quantitative and qualitative results for all the methods under study.

4.4.1 Datasets and Parameter Settings

We performed experimental validation on the following datasets, including CMU PIE face dataset [Sim et al., 2002], Car dataset [Krause et al., 2013], Cat dataset [Zhang et al., 2008], Gore face dataset [Peng et al., 2012] and the Multiview dataset [Hirschmüller and Scharstein, 2007]¹. For each dataset, we randomly selected half of the images to form a category-specific dataset and between 10 and 15 images from the remaining half as ground-truth images for denoising. It is to be noted here that we have disjoint image sets for the test and training *i.e.* neither the same people appear in the test and training images, nor the same objects with different scale and pose. To generate noisy images, we corrupt the test images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 30, 50, 70, 100$, similar to the practice employed elsewhere [Yue et al., 2014; F. Chen and Yu, 2015; Luo et al., 2015; Xu et al., 2015a; Yue et al., 2015]. We also intend to demonstrate the effectiveness of our algorithm at the high noise std of 50 and beyond.

For evaluation purposes, we use Peak Signal-to-Noise Ratio (PSNR) index as the error metric. We compare our proposed method with numerous state-of-the-art methods, including BM3D [Dabov et al., 2007b], WNNM [Gu et al., 2014], NLM [Buades et al., 2005], SAPCA [Dabov et al., 2009], TSID [Zhang et al., 2010], EPLL [Zoran and Weiss, 2011], PCLR [F. Chen and Yu, 2015], PGPD [Xu et al., 2015a] and TID [Luo et al., 2015]. To ensure a fair comparison, we modify the state-of-the-art internal denoising methods of NLM, BM3D, SAPCA, and TSID to perform the search on class-specific image datasets. We use the same settings as their original implementations.

Our method shares a number of common parameters with algorithms that exploit patch similarity and inherits the parameter values from the prior works. Similar to BM3D [Dabov et al., 2007b] and WNNM [Gu et al., 2014], we choose a patch size of 8. When searching for support patches, we select $L = 16$ candidate images that are most similar to the noisy image, as described in the denoising method using targeted

¹In practice, we can utilize images of particular object categories from publicly available datasets such as PASCAL VOC and ImageNet.

databases, *i.e.* TID [Luo et al., 2015]. Besides, we employ a search window with a size of 51×51 in each candidate image. In the last stage of support patch search, the number of nearest neighbors k is set to 16, similar to the external denoising methods of eBM3D, eSAPCA, and eNLM.

Further, we set the parameters specific to our optimization problem as follows, $\lambda_0 = 1$, $\lambda_1 = 0.5$, $\lambda_2 = 10$ and $\rho = 0.18$. The values of λ_0 , λ_1 and λ_2 are determined by a sensitivity analysis such as that in Section 4.4.4, using a small number of noisy images as the validation set. Our algorithm inherits the value of ρ from the PCLR, WNNM, and PGPD methods.

4.4.2 Influence of the External Dataset Size

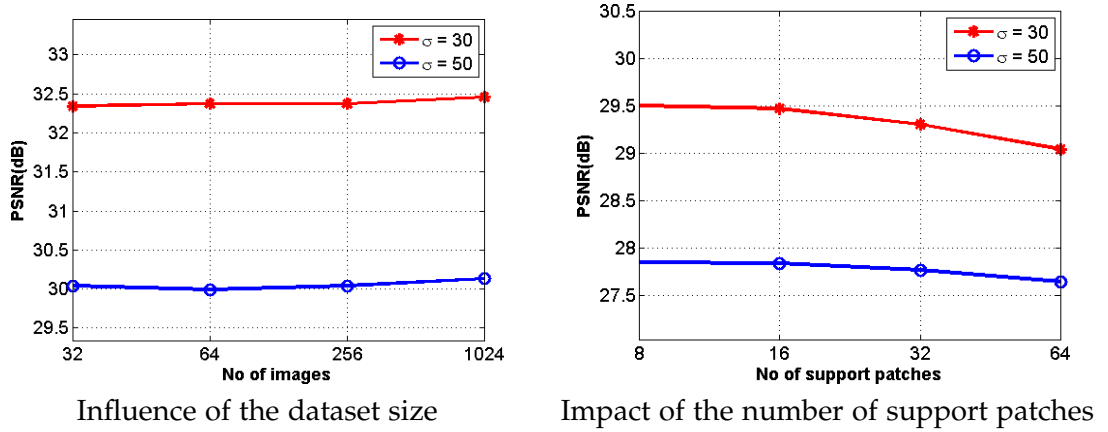


Figure 4.3: Denoising accuracy (in PSNR) at noise standard deviations $\sigma_n = 30$ and $\sigma_n = 50$. Left: Our method is robust to the changes in the dataset size, which has a low impact on the results. Right: Increasing the number of support patches slightly degrades the denoising results.

Table 4.1: Run-time comparisons (in seconds) on a test image of size 304×228 .

Method	BM3D	eBM3D	eNLM	eSAPCA	eTSID	Ours
Time (s)	1.12	173.5	164.6	178.8	178.9	119.9
Method	PCLR	PGPD	TID	EPLL	WNNM	Ours
Time (s)	192.4	12.5	172.2	39.9	211.8	119.9

Here, we examine the influence of the size of the external dataset on the denoising performance, while fixing all the other parameters. To this end, we experiment with dataset sizes of 32, 64, 256 and 1024 by incrementally adding images to the clean image dataset. We choose 15 images among those, not in the dataset and simulate

Table 4.2: Denoising performance (in PSNR) when using different image category datasets. The PSNR is maximal when the external dataset category matches the noisy image category.

	Dataset				
Noisy	Face	Cat	Texture	Text	Car
Face	26.80	24.79	22.79	17.03	24.89
Cat	25.01	28.00	25.24	19.57	25.87
Texture	24.09	24.67	28.13	19.33	24.62
Text	8.43	15.41	14.50	21.09	16.33
Car	18.41	19.80	18.85	16.93	21.90

noisy input by adding Gaussian noise with a standard deviation σ_n of 30 and 50. The left panel in Figure 4.3 demonstrates the robustness of our algorithm to the dataset size, showing that an increasing dataset size only slightly improves the denoising accuracy. Even with a small dataset size of 32, our algorithm can achieve an average PSNR of 32.3 dB for $\sigma_n = 30$ and 30 dB for $\sigma_n = 50$.

4.4.3 Influence of the number of support patches

Similarly, we test the robustness of our algorithm to the number of support patches required for denoising a single patch. For this purpose, we use 8, 16, 32 and 64 support patches per noisy patch. The plots on the right-hand side of Figure 4.3 shows that the average PSNR declines as the number of support patches increases. The main reason for this phenomenon is that the variation in appearance between the support patches is likely to increase with a larger number of support patches, and their aggregation would result in a loss of local details due to averaging.

4.4.4 Relative Importance of the Priors

We assess the relative contribution of the Gaussian prior and the low-rank term on the Gore dataset for $\sigma = 50$. The presence of both terms improves the PSNR compared to the scenario where one is absent. For example, when $\lambda_1 \in \{1, 10, 40\}$ and $\lambda_2 = 0$ the resulting PSNR are $\{27.58, 27.56, 27.56\}$, respectively. When $\lambda_1 = 0$ and $\lambda_2 \in \{1, 10, 40\}$ the results are $\{20.14, 26.33, 27.09\}$. When $\lambda_1 = 1$ and $\lambda_2 = 10$, the average PSNR increases to **27.82** dB.

4.4.5 Runtime Comparisons

We have implemented our algorithm in MATLAB on an Intel CoreTM i7 machine with 16 GB of memory. In Table 4.1, we show the running times for various methods

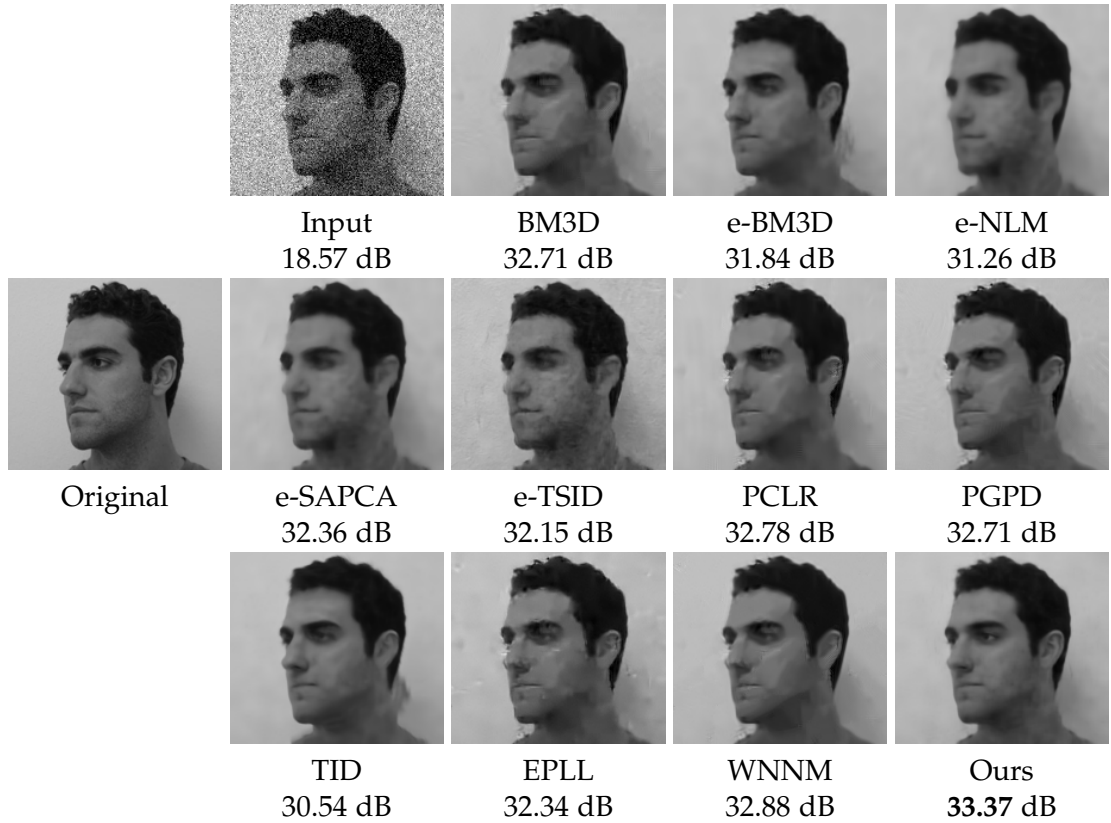


Figure 4.4: Denoising results produced by different methods for a face image in a profile view from the FEI face dataset [Thomaz and Giraldi, 2010] when $\sigma_n = 30$. Our method can denoise the input image even with a different pose from those in the noise-free dataset (Differences are better viewed with high-resolution display).

including ours for an image of size 304×228 and an external dataset containing 10 images of similar dimensions. The running time of our method, *i.e.* 119s is shorter than the MATLAB implementations of various state of the art external image denoising methods *e.g.*, eNLM, eBM3D, eTSID, TID, WNNM, and eSAPCA. We observe that our method spends most of its time on patch search. The speed of our algorithm can be improved by applying fast patch search algorithms *e.g.*, KD tree [Muja and Lowe, 2014] and patch match [Mahmoudi and Sapiro, 2005; Vignesh et al., 2010]. In addition, GPU implementations can be employed to parallelize the denoising of patches in independent threads.

4.4.6 Role of the External Image Category

Now we illustrate the importance of choosing the correct external image category for denoising. To this end, we provide datasets of different object categories as input

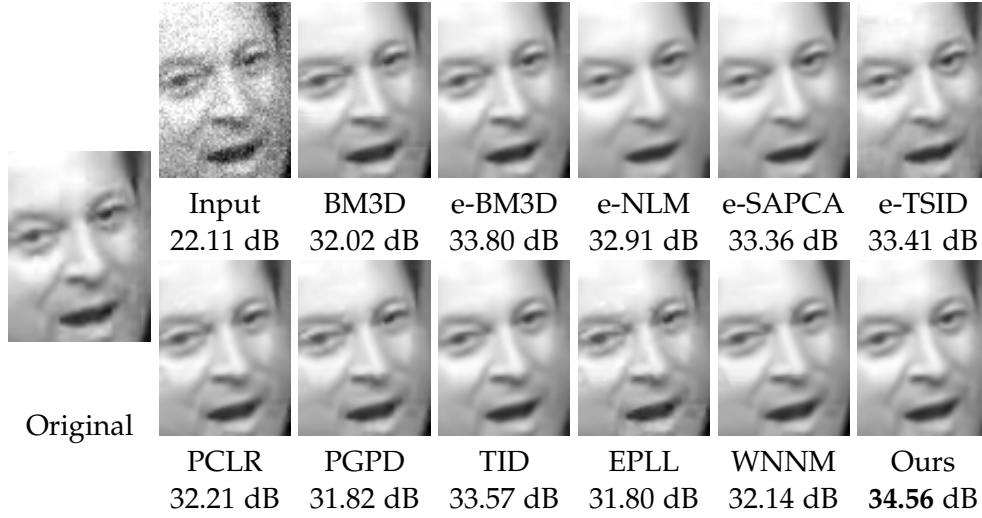


Figure 4.5: Denoising results produced by different methods for a face image selected from the Gore dataset [Peng et al., 2012] when $\sigma_n = 20$. Our method is able to denoise the face image even with a different pose from those in the noise-free dataset.

Table 4.3: Performance comparison between our method and internal denoising techniques on several datasets, in terms of PSNR (in dB).

$\sigma_n = 30$											
	BM3D	NLM	SAIST	SAPCA	TSID	DDID	PID	NLB	WNNM	AID	Ours
Gore	29.28	27.59	29.77	29.00	28.21	29.23	29.21	29.04	29.35	29.32	29.95
Cat	30.08	27.08	29.86	30.03	29.45	29.97	29.95	29.86	30.11	29.96	31.18
CMU	32.56	30.37	32.48	32.61	31.92	32.65	32.73	32.33	32.63	32.45	33.38
View	28.35	27.08	28.20	28.43	28.82	28.19	28.34	28.26	28.46	28.31	31.96
$\sigma_n = 50$											
Gore	26.54	23.54	27.11	26.23	25.37	26.48	26.57	26.41	26.37	26.58	27.82
Cat	27.96	24.69	27.76	27.91	27.18	27.89	27.91	27.70	27.97	27.80	28.79
CMU	30.17	27.92	30.07	30.11	29.29	30.21	30.48	29.84	30.21	29.90	30.64
View	26.35	24.69	26.10	26.39	26.53	26.17	26.32	26.15	26.39	26.27	28.64
$\sigma_n = 70$											
Gore	24.94	21.30	24.01	25.13	23.43	24.68	24.88	24.64	24.32	24.81	25.58
Cat	26.56	23.21	26.24	26.21	25.64	26.45	26.59	26.15	26.52	26.36	26.80
CMU	28.45	26.00	28.36	27.90	27.52	28.51	28.89	27.95	28.58	28.16	28.72
View	25.14	23.21	24.94	24.92	25.09	24.92	25.07	24.83	25.16	25.00	27.16
$\sigma_n = 100$											
Gore	23.21	19.06	22.21	23.31	21.30	22.77	23.05	22.76	22.48	22.95	23.86
Cat	25.08	21.95	24.69	24.56	23.97	24.91	25.20	24.46	25.02	24.87	25.21
CMU	26.57	24.32	26.56	25.90	25.62	26.63	27.14	26.10	26.74	26.36	26.59
View	23.89	21.95	23.50	23.57	23.64	23.62	23.86	23.49	23.85	23.74	24.79

Table 4.4: Performance comparison between our method and external denoising techniques on several datasets, in terms of PSNR (in dB).

$\sigma_n = 30$									
	eBM3D	eNLM	eSAPCA	eTSID	PCLR	PGPD	TID	EPLL	Ours
Gore	29.49	27.30	28.56	29.19	29.04	29.38	29.68	29.21	29.95
Cat	28.35	24.30	26.01	26.98	30.11	30.07	26.21	29.77	31.18
CMU	31.59	29.63	30.74	30.65	32.66	32.55	28.56	32.59	33.38
View	26.79	25.21	26.13	24.45	28.37	28.34	23.90	28.17	31.96
$\sigma_n = 50$									
Gore	26.95	26.21	26.79	26.64	25.80	26.70	27.55	26.59	27.82
Cat	26.42	23.57	25.07	25.68	27.87	28.01	24.77	27.74	28.79
CMU	29.27	28.12	29.06	28.35	30.26	30.18	27.12	29.51	30.64
View	24.88	24.47	24.71	23.42	26.32	26.39	23.01	26.13	28.64
$\sigma_n = 70$									
Gore	25.42	25.11	25.37	24.68	23.70	24.89	25.40	24.69	25.58
Cat	25.13	23.23	24.08	24.16	26.48	26.63	23.34	26.23	26.80
CMU	27.68	27.31	27.63	26.23	28.66	28.57	26.04	27.86	28.72
View	23.55	23.54	23.81	23.23	25.05	25.21	22.19	24.86	27.16
$\sigma_n = 100$									
Gore	23.38	23.26	23.32	22.13	21.93	23.00	23.30	22.85	23.86
Cat	23.90	22.20	23.10	22.44	25.09	25.12	23.06	24.82	25.21
CMU	25.88	25.65	25.95	23.84	26.91	26.71	24.49	26.13	26.59
View	22.28	22.67	21.84	22.63	23.80	23.93	21.25	23.61	24.79

Table 4.5: Denoising performance in PNSR (dB) on color images for noise levels $\sigma_n = 30, 50, 70, 80, 100$. Best results are in bold.

Datasets	Methods	σ_n				
		30	50	70	80	100
FEI	CBM3D	35.59	33.23	31.55	30.88	29.68
	NLB	34.99	33.32	31.60	30.91	29.65
	Ours	35.99	33.66	32.30	31.62	30.39
Views	CBM3D	29.28	27.19	25.94	25.46	24.60
	NLB	29.06	27.12	25.44	24.95	24.02
	Ours	30.07	27.75	26.94	26.50	25.65
CMU	CBM3D	31.70	29.29	27.68	27.05	25.86
	NLB	32.56	29.11	25.72	24.41	22.23
	Ours	32.84	30.90	29.56	29.15	28.30

to our method for denoising the same noisy images. The categories involved in our experiment are Face (Gore dataset), Cat, Texture (from the Multiview dataset),

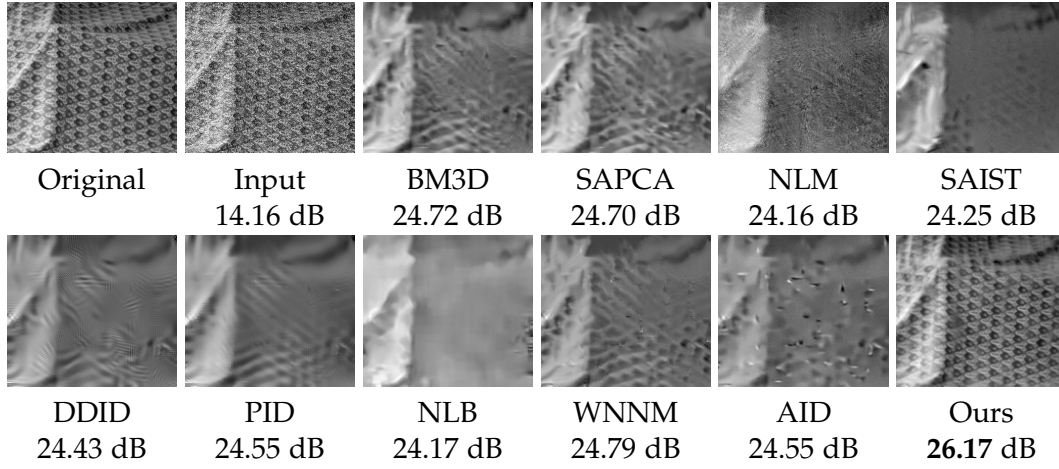


Figure 4.6: Visual denoising results produced for $\sigma_n = 50$, by several methods for a sample texture image from the Multiview dataset [Hirschmüller and Scharstein, 2007]. Our approach can recover much more texture details as compared to competing methods.

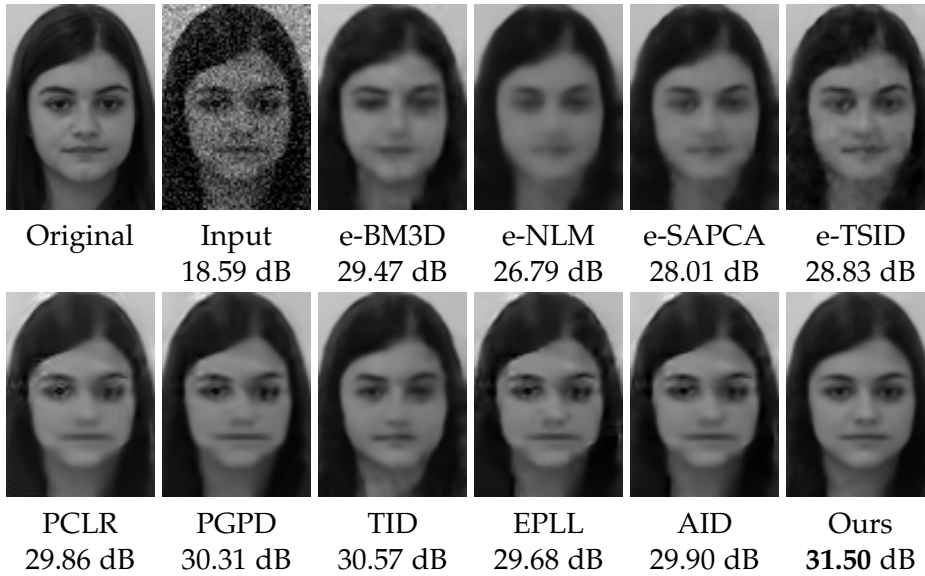


Figure 4.7: Denoising results achieved by various methods for a sample image with a noise standard deviation $\sigma_n = 30$. The ground truth image is from the Gore dataset [Peng et al., 2012].

Text [Luo et al., 2015] and Car. In Table 4.2, we show the average PSNR of the denoised images for each pair of noisy image category and dataset category. Note that the PSNR values reported are averaged across all the mentioned noise levels ($\sigma_n = 30, 50, 70, 100$) and noisy images. Each row of the table corresponds to a noisy

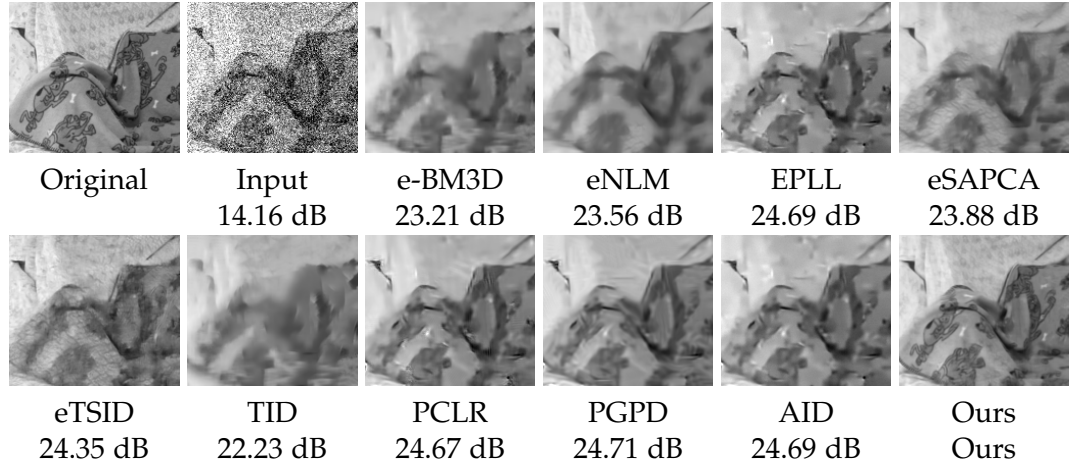


Figure 4.8: Visual denoising results for a texture image selected from the Multiview dataset [Hirschmüller and Scharstein, 2007] where $\sigma_n = 50$. Our method can recover much more texture details than the others (please zoom-in to see details).

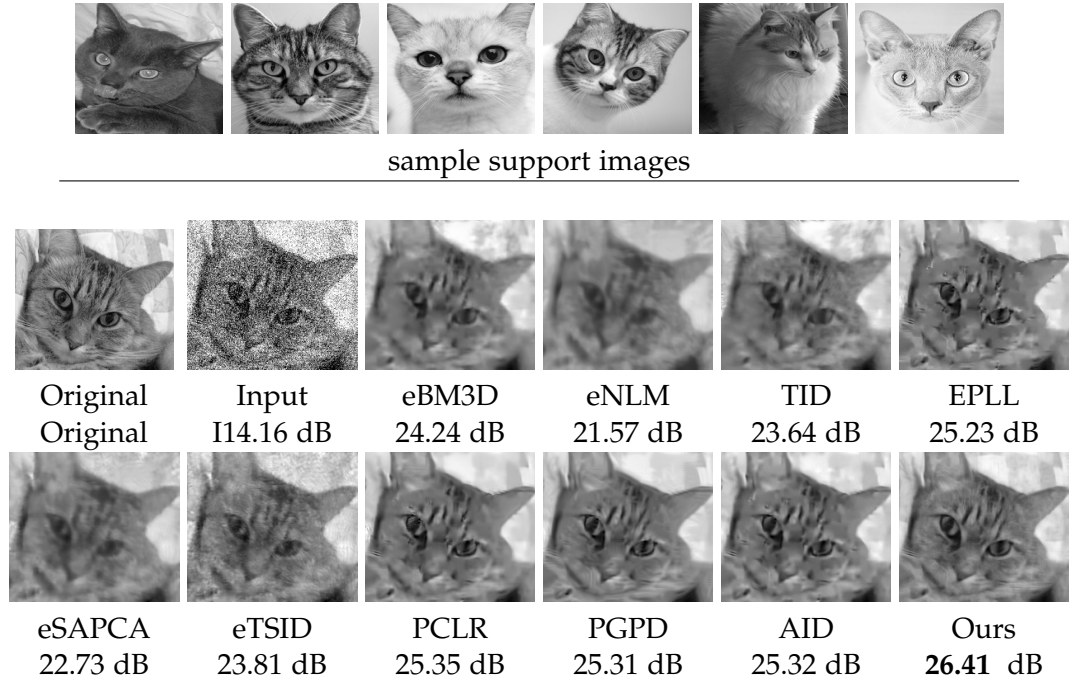


Figure 4.9: Denoising results for different methods from the dataset in [Zhang et al., 2008] when $\sigma_n = 50$. The top two rows show the candidate images from the dataset that are most similar to the noisy image.

image category while each column represents a dataset category.

The overall trend is that the PSNR for each noisy image category reaches its max-

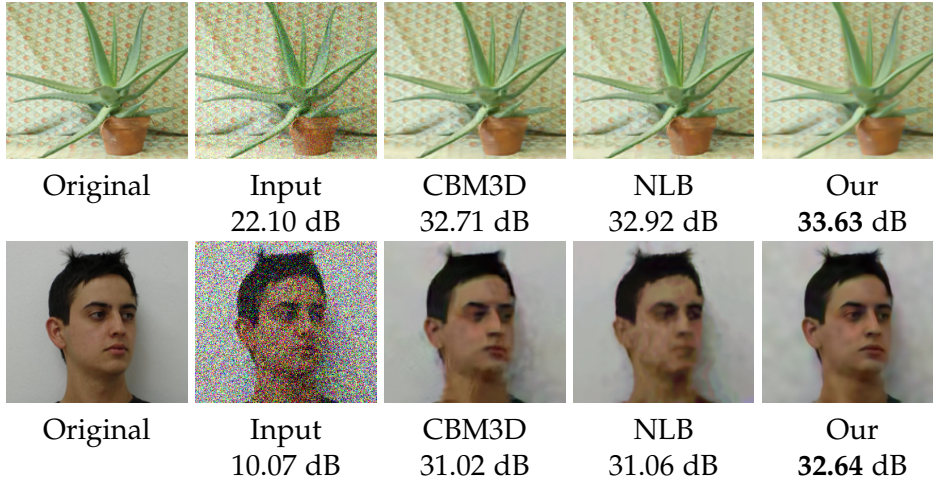


Figure 4.10: Comparison of a few denoising methods on color images from the datasets in [Hirschmüller and Scharstein, 2007] and [Thomaz and Giraldi, 2010], where the noise standard deviations are $\sigma_n = 20$ and $\sigma_n = 80$, respectively. Our method can recover much more details than the others.

imum when the dataset belongs to the same category, as can be observed along the diagonal of Table 4.2. On the other hand, the PSNR diminishes significantly when the dataset belongs to a different category, which confirms the benefit of category-specific information for denoising purposes. This observation also demonstrates the ability of our algorithm to extract useful category-specific information from the support patches.

4.4.7 Sensitivity to Pose Variations

We analyze the response of our method when the noisy test images contain significant pose variations including out-of-plane rotations, semi-profile views, and different camera views. We present sample results in Figures 4.4 and 4.5.

As can be seen, our method attains qualitatively the most appealing results and quantitatively the best PSNR scores among the all considered methods thanks to its efficient scheme of support patch selection. It restores the different pose faces without visible deterioration of the facial details. Meanwhile, the competitive methods generate clearly visible artifacts induced by the noise distribution.

4.4.8 Comparisons with Internal Denoising Methods

We first present the quantitative comparisons with the state-of-the-art internal denoising methods in Table 4.3. The scores are averaged across all test images in the

datasets. Overall, our method is the best performer.

In Figure 4.6 it is visible that the proposed algorithms can restore high-frequency details with a closer resemblance to the ground truth than the existing internal denoising methods. Specifically, the highly-textured pattern is reproduced by our method, while these details are highly distorted or smoothed out by the other methods. Upon close inspection, most of the other methods either smooth out the periodical variations of the background texture or introduce additional artifacts and artificial textures. This phenomenon explains the much inferior PSNR produced by the other methods.

4.4.9 Comparisons with External Denoising Methods

In Table 4.4, we present the average PSNR measured across the Gore, Cat, CMU-PIE, and Multiview datasets. Among the considered methods, ours is the best overall performer (in terms of PSNR) across most combinations of datasets and noise levels.

In addition to the superior quantitative results, our method also delivers superior visual quality. As an example, we provide visual comparisons between the results generated by our method and the state of the art alternatives for a face image with the noise level $\sigma_n = 30$ and a texture image with the noise level $\sigma_n = 50$, as shown in Figures 4.7 and 4.8, respectively.

In Figure 4.7, the face image denoised by our method is indeed of higher visual quality than their counterparts. Within the face region, our algorithm can reproduce all the facial parts without distortion, whereas the other methods cause different kinds of artifacts. Furthermore, most of the other methods in this comparison introduce visible artifacts on the forehead and the chin. The lower performance of the other methods could be explained by their difficulty in finding correct matches for patch grouping due to high noise and high variance within each patch group. As a result of inaccurate grouping of patches, texture details are destroyed, and incoherent patterns are generated.

In Figure 4.8, the proposed algorithms are able to restore high-frequency details with a closer resemblance to the ground truth than existing methods. Specifically, the highly-textured pattern is reproduced by our method, while these details are highly distorted or smoothed out by the other methods. Upon close inspection, most of the other methods either smooth out the periodical variation of the background texture or introduce additional artifacts and artificial textures. This phenomenon implies a much inferior PSNR produced by others than our method.

4.4.10 Robustness to Misalignments and Rotations

In the top row of Figure 4.9, we show sample noise-free images in the cat database. In the bottom row, we show a noisy input image and the corresponding denoised one. We observe that while the appearances and expressions of cats in the support images significantly vary, and there are severe misalignments between them, our method still generates much higher PSNR than other methods.

4.4.11 Extension to Color Images

For noisy color images, we first perform a luminance-chrominance² transformation. Let Y denotes the luminance channel, and U and V denote the chrominance channels. Often, the luminance channel provides prominent texture information while the chrominance channels endure lower SNR [Pirinen et al., 2007].

We specifically deal with the high noise variance in the Y channel with our method, while simply applying BM3D to the chrominance channels. In Table 4.5 and Fig. 4.10, we present comparison with the current state-of-the-art color image denoising algorithms [Lebrun et al., 2013; Dabov et al., 2007a]. One can observe that our method outperforms all existing methods on three benchmark datasets for five different noise levels.

4.5 Discussion

We have presented an effective algorithm for denoising object images using support patches from an image dataset of the same category. The patch selection strategy aims to draw support patches within a locality of the input patch from the best candidate images. The key difference from existing external denoising methods is the formulation of the denoising problem in a transform domain. Also, we include novel terms to model support patch group membership and to promote the similarity between the noisy and the support patches.

A critical feature to discuss is the sensitivity of the algorithm specifically, to changes in font size (for text dataset), object size (cat dataset), facial expressions (face dataset) and viewing angle (multi-view dataset). Similarly, automatic selection of patch size and the number of patches required for the denoising will help improve the performance. Furthermore, we have validated the robustness of our algorithm to the dataset size, the number of support patches, and verified the importance of choosing the appropriate dataset category. Overall, our algorithm outperforms all state-of-the-art methods included in our study, both numerically and visually. We

²We consider opponent color models, yet any other transformation such as $YCbCr$, Lab can be used.

have shown that our approach outperforms both internal and external denoising algorithms and provide ample examples to demonstrate the superior performance.

Chaining Identity Mapping Modules for Image Denoising

You don't take a photograph, you make it.

Ansel Adams

In this chapter, we continue our investigation of the image denoising problem and propose to learn a fully-convolutional network model that consists of a Chain of Identity Mapping Modules (CIMM) for this task. The CIMM structure possesses two distinctive features that are important for the noise removal task. Firstly, each residual unit employs identity mappings as the skip connections and receives pre-activated input in order to preserve the gradient magnitude propagated in both the forward and backward directions. Secondly, by utilizing dilated kernels for the convolution layers in the residual branch, in other words within an identity mapping module, each neuron in the last convolution layer can observe the full receptive field of the first layer. After being trained on the BSD400 dataset [Martin et al., 2001], the proposed network produces remarkably higher numerical accuracy and better visual image quality than the state-of-the-art when being evaluated on conventional benchmark images and the BSD68 dataset [Roth and Black, 2009].

5.1 Introduction

As mentioned in the earlier chapters, image denoising is an essential building module for various algorithms. In the past few years, the research focus in this area has been shifted to how to make the best use of image priors. To this end, several approaches attempted to exploit non-local self similar (NSS) patterns [Buades et al., 2005; Dabov et al., 2007b, 2009], sparse models [Gu et al., 2014; Peng et al.,

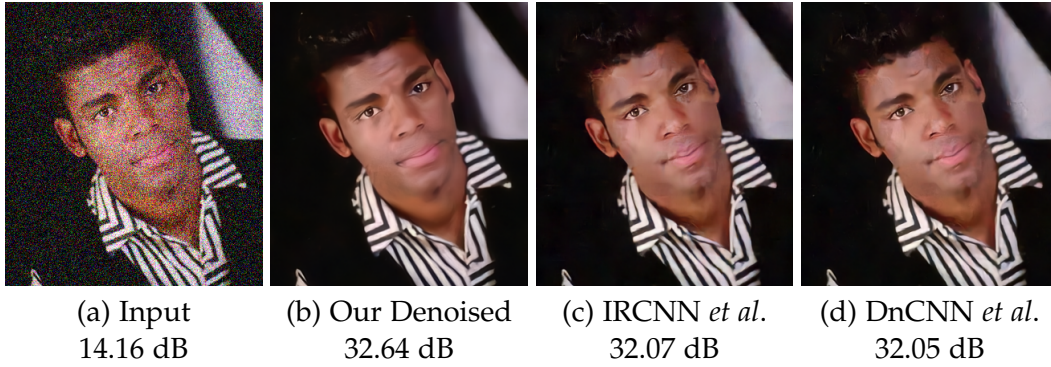


Figure 5.1: Denoising results for an image corrupted by the Gaussian noise with $\sigma = 50$. Our result has the best PSNR score, and unlike other methods, it does not have over-smoothing or over-contrasting artifacts. Best viewed in color on high-res display.

2012], gradient models [Osher et al., 2005; Xu and Osher, 2007; Weiss and Freeman, 2007], Markov random field (MRF) models [Roth and Black, 2009], external denoising [Yue et al., 2014; Anwar et al., 2017c; Luo et al., 2015] and convolutional neural networks [Zhang et al., 2017a; Lefkimmatis, 2016; Zhang et al., 2017b].

In most of the computer vision and image algorithms convolutional neural network is leading in terms of performance. This is due to the availability of large datasets and the capability to learn filters and relationships between input and output that were hand engineered in traditional algorithms. In traditional algorithms, the aim was to reduce the noise through filtering or thresholding; however, it is simple to learn the residue (noise) through CNN and then can be easily removed from the image.

Current convolutional neural network based image denoising methods [Burger et al., 2012; Zhang et al., 2017a,b; Lefkimmatis, 2016] connect weight layers consecutively and learn the mapping by brute force without putting any effort into the architecture. One problem with such an architecture is the addition of more weight layers to increase the depth of the network. Even if the new weight layers are added to the mentioned CNN based denoising methods, it will fall into gradients vanishing problem and impel it further [Bengio et al., 1994]. This property of increasing the size of the network is important and helps in performance boost [Lim et al., 2017; He et al., 2016a]. Similarly, the current CNN methods require hyper-parameter settings, extensive fine-tuning, stage-wise training, embedding non-local self similar priors or predicting noise without knowing the underlying image structure. Therefore, our goal is to propose a model that overcomes these deficiencies.

Another reason is the lack of true color denoising as most of the current de-

noising systems are either for grayscale image denoising or treat each color channel separately ignoring the relationship between the color channels. Only a handful of works [Dabov et al., 2007a; Anwar et al., 2017c; Zhang et al., 2017a; Lefkimmiatis, 2016] approached color image denoising in its own context.

To provide a solution, our choice is the convolutional neural networks in a discriminative prior setting for image denoising. There are many advantages of using CNNs, including efficient inference, incorporation of robust priors, integration of local and global receptive fields, regressing on nonlinear models, and discriminative learning capability. Furthermore, we propose a modular network where we call each module as a mapping modules (MM). The mapping modules can be replicated and easily extended to any arbitrary depth for performance enhancement.

Contributions: The major contributions of this work can be summarized as follows:

- An effective CNN architecture that consists of a Chain of Identity Mapping modules (CIMM) for image denoising. These modules share a common composition of layers, with residual connections between them to facilitate training stability.
- The use of dilated convolutions for learning suitable filters for denoising image patches at different levels of spatial extent.
- A single denoising network that can handle various noise levels.
- Increasing efficiency by reducing the number of layers while increasing the dilation to preserve accuracy.

The rest of the chapter is organized as follows. First, we introduce the architecture of the network in section 5.2, followed by experimental section 5.3. In the last section 5.4, we discuss the main features of this chapter and provide a summary of the chapter.

5.2 Chain of Identity Mapping Modules

This section presents our approach to image denoising by learning a Convolutional Neural Network consisting of a Chain of Identity Mapping Modules (CIMM). Each module is composed of a series of pre-activation units followed by convolution functions, with residual connections between them. Section 5.2.2 formulates the learning objective. Subsequently, the meta-structure of the CIMM network in Section 5.2.1.

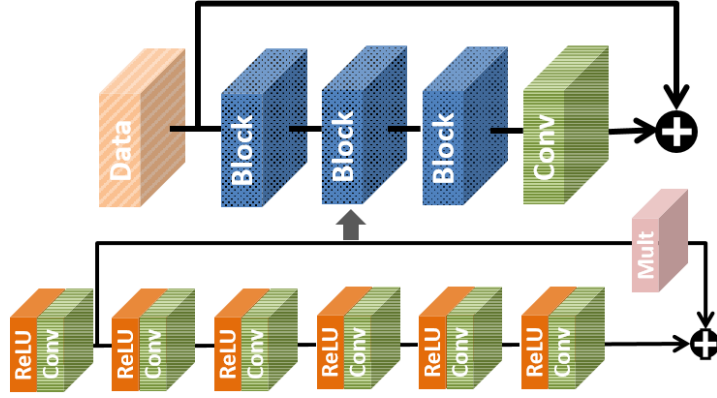


Figure 5.2: The proposed network architecture, which consists of multiple modules with similar structures. Each module is composed of a series of pre-activation-convolution layer pairs.

5.2.1 Network Design

Residual learning has recently delivered state of the art results for object classification [He et al., 2015a, 2016b] and detection [Lin et al., 2016], while offers training stability. Inspired by the Residual Network variant with identity mapping [He et al., 2016b], we adopt a modular design for our denoising network. The design consists of a Chain of Identity Mapping modules (CIMM).

5.2.1.1 Network Elements

Figure 5.2 depicts the entire architecture, where identity mapping modules are shown as a dotted blue blocks, which are in turn composed of basic rectified linear unit “ReLU” (orange) and convolution (horizontal stripes of green) layers. The output of each module is a summation of the identity function and the residual function. In our experiments, we typically employ 64 filters of size 3×3 in each convolution layer.

The meta-level structure of the network is governed by three parameters: the number of identity modules (*i.e.* \mathbf{M}), the number of pre-activation-convolution pairs in each module (*i.e.* \mathbf{L}), and the number of output channels (*i.e.* \mathbf{C}), which we fixed across all the convolution layers.

The high-level structure of the network can be viewed as a chain of identity mapping modules, where the output of each module is fed directly into the subsequent one. Subsequently, the output of this chain is fed to a final convolution layer to produce a tensor with the same number of channels as the input image. At this point, the final convolution layer directly predicts the noise component from a noisy image.

The noisy image/patch is then subtracted from the input to recover the noise-free image .

The identity mapping modules are the building blocks of the network, which share the following structure. Each module consists of two branches: a residual branch and an identity mapping branch. The residual branch of each module contains a series of layers pairs, *i.e.* a nonlinear pre-activation (typically ReLU) layer, followed by a convolution layer. Its main responsibility is to learn a set of convolution filters to predict image noise. In addition, the identity mapping branch in each module allows the propagation of the loss gradients in both directions without any bottleneck.

5.2.1.2 Justification of the Design

For image denoising, several previous works have adopted a fully convolutional network design, without any pooling mechanism [Zhang et al., 2017a; Kim et al., 2016b; Zhang et al., 2017b]. This is necessary in order to preserve the spatial resolution of the input tensor across different layers. We follow this design by using only non-linear activations and convolution layers across our network.

Furthermore, as inspired by non-local denoising method, we design the convolution layers in such a way that neurons in the last layer of each identity mapping (IM) module observe the full spatial receptive field in the first convolution layer. This design helps learning to connect input neurons at all spatial locations to the output neurons, in much the same way as well-known non-local mean methods such as [Dabov et al., 2007b; Buades et al., 2005]. Instead of using a unit stride within each layer, we also experimented with dilated convolutions to increase the receptive fields of the convolution layers. By this design, we can reduce the depth of each IM module while the final layer’s neurons can still observe the full input spatial extent. We provide further details on the relationship between network depth and kernel dilation in section 5.3.5.3.

Pre-activation has been shown to offer the highest performance for classification when used together with identity mapping [He et al., 2016b]. In a similar fashion, our design employs ReLU before each convolution layer. This design differs from existing neural network architectures for denosing [Kim et al., 2016b; Zhang et al., 2017a; Lefkimmiatis, 2016]. The pre-activation helps training to converge more easily, by while the identity function preserves the range of gradient magnitudes. Also, the resulting network generalizes better as compared to the post-activation alternative. This property enhances the denoising ability of our network.

5.2.1.3 Our Formulation

Now we formulate the prediction output of this network structure for a given input patch \mathbf{y} . Let \mathcal{W} denote the set of all the network parameters, which consists of the weights and biases of all constituting convolution layers. Specifically, we let $w_{m,l}$ denote both the kernel and bias parameters of the l -th convolution layer in the residual branch of the m -th module.

Within such a branch, the intermediate output of the l -th ReLU-convolution pair and of the m -th module is a composition of two functions

$$\mathbf{z}_{m,l} = f(g(\mathbf{y}_{m,l}); w_{m,l}), \quad (5.1)$$

where f and g are the notation for the convolution and the ReLU functions, $\mathbf{z}_{m,l}$ is the output of the l -th ReLU-convolution pair of m -th module. By composing the series of ReLU-convolution pairs, we obtain the output of the m -th residual branch as

$$\mathbf{r}_m = -\mathbf{z}_{m,0} + f(g(\dots f(g(\mathbf{y}_{m,0}; w_{m,0})) \dots); w_{m,l}), \quad (5.2)$$

where $\mathbf{z}_{m,0}$ is the output of the first ReLU-convolution pair. Chaining all the identity mapping modules, we obtain the output as $\sum_{m=1}^M \mathbf{r}_m$. Finally, the output of this chain is convolved with a final convolution layer with learnable parameters w_{m+1} to predict the noise component as $h(\mathbf{y}, \mathcal{W}) = f(\mathbf{y} + \sum_{m=1}^M \mathbf{r}_m, w_{m+1})$.

5.2.2 Learning to Denoise

Our convolutional neural network (CNN) is trained on image patches or regions rather than at the image-level. This decision is driven by a number of reasons. Firstly, it offers random sampling of a large number of training samples at different locations from various images. Random shuffling of training samples is well-known to be a useful technique to stabilize the training of deep neural networks. Therefore, it is preferable to batch training patches with a random, diverse mixture of local structures, patterns, shapes and colors. Secondly, there has been success in approaches that learn image patch priors from external data for image denoising [Zoran and Weiss, 2011].

From a set of noise-free training images, we randomly crop a number of training patches $\mathbf{x}_i, i = 1, \dots, N$ as the groundtruth. The noisy version of these patches is obtained by adding (Gaussian) noise to the ground truth training images. Let us denote the set of noisy patches corresponding to the former as $\mathbf{y}_i, i = 1, \dots, N$. With this setup, our image denoising network (described in Section 5.2.1) is aimed to reconstruct a patch $\mathbf{x}_i^* = h(\mathbf{y}_i, \mathcal{W})$ from the input patch \mathbf{y}_i .

The learning objective is to minimize the following sum of squares of ℓ_2 -norms

$$\mathcal{L} \triangleq \frac{1}{N} \sum_{i=1}^N \|h(\mathbf{y}_i, \mathcal{W}) - \mathbf{x}_i\|^2. \quad (5.3)$$

To train the proposed network on a large dataset, we minimize the objective function in Equation 5.3 on mini-batches of training examples. Training details for our experiments are described in Section 5.3.2.

5.3 Experiments

5.3.1 Benchmark Datasets and Baseline Methods

We performed experimental validation on the widely used classical images (same number and images as [Zhang et al., 2017a]). Similarly, we also use DnD datasets [Plötz and Roth, 2017] consists of real 1000 images and BSD68 dataset [Roth and Black, 2009] composed of 68 images. It is to be noted here, that our BSD400 dataset [Martin et al., 2001] for training and BSD68 dataset [Roth and Black, 2009] for testing are disjoint. To generate noisy test images, we corrupt the images by additive white Gaussian noise with standard deviations (std) of $\sigma_n = 15, 25, 50, 70$, as employed by [Zhang et al., 2017b,a; Lefkimmiatis, 2016]. For evaluation purposes, we use the Peak Signal-to-Noise Ratio (PSNR) index as the error metric. We compare our proposed method with numerous state-of-the-art methods, including BM3D [Dabov et al., 2007b], WNNM [Gu et al., 2014], MLP [Burger et al., 2012], EPLL [Zoran and Weiss, 2011], TNRD [Chen and Pock, 2017], IRCNN [Zhang et al., 2017b], DnCNN [Zhang et al., 2017a] and NLNET [Lefkimmiatis, 2016]. To ensure a fair comparison, we use the default setting provided by the respective authors. In all experiments the input image is grayscale except those in sections 5.3.6.3 and 5.3.6.4.

5.3.2 Training Details

The training input to our network is noisy and noise-free patch pairs of size 40×40 cropped randomly from the BSD400 dataset [Martin et al., 2001]. Note that there is no overlap between the training and evaluation datasets. We also augment the training data with horizontally and vertically flipped versions of the original patches and those rotated at an angle of $\frac{\pi n}{2}$, where $n = 1, 2, 3$. The training patches are randomly cropped on the fly from the 400 images of BSD400 dataset.

We offer two strategies for handling different noise levels. The first one is to train a network for each specific noise level and we call model as noise-specific model. Alternatively, we train a single model for the noise range $[1, 50]$ (similar to [Zhang

et al., 2017a]) and we refer to this model as noise-agnostic model. At each update of training, we construct a batch by randomly selecting noisy patches with noise levels between 1 and 50.

We implement the denoising method in the Caffe framework on two Tesla P100 GPUs, and employ the Adam optimization algorithm [Kingma and Ba, 2014] for training. The initial learning rate was set to 10^{-4} and the momentum parameter was 0.9. We scheduled the learning rate such that it is halved at every 1.5×10^5 mini-batches of size 64. We train our network from scratch by a random initialization of the convolution weights according to the method in [He et al., 2015b] and a regularization strength, *i.e.* weight decay, of 0.0001.

5.3.3 Boosting Denoising Performance

To boost the performance of the trained model, we use the late fusion/geometric transform strategy as adopted by [Timofte et al., 2016]. During the evaluation, we perform eight types of augmentation (including identity) of the input noisy images y as $y_i^t = \Gamma_i(y)$ where $i = 1, \dots, 8$. From these geometrically transformed images, we estimate corresponding denoised images $\{\hat{x}_1^t, \hat{x}_2^t, \dots, \hat{x}_8^t\}$, where $\hat{x}_i^t = h(\hat{y}_i^t, W)$ using our model. To generate the final denoised image \hat{x} , we perform the corresponding inverse geometric transform $\tilde{x}_i^{-t} = \Gamma_i^{-1}(\hat{x}_i^t)$ and then take the average of the outputs as $\tilde{x} = \frac{1}{8} \sum_{i=1}^8 \tilde{x}_i^t$. This strategy is beneficial as it saves training time and have small number of parameters as compared to individually trained eight models. We also found empirically that this fusion method gives approximately the same performance as the models trained individually with geometric transform.

5.3.4 Identity Mapping Modules

The structure of the mapping modules used in our experiments is depicted in Table 5.1. Each module consists of a series of ReLU + Conv pair. All the convolution layers have a kernel size of 3×3 and 64 output channels. The kernel dilation and padding are same in each layer and vary between 1 and 3. The skip connection connects the output of the first pair of ReLU + Conv to the last pair ReLU + Conv as shown in Figure 5.2

5.3.5 Ablation Study

5.3.5.1 Influence of the Patch Size

In this section, we show the role of the patch size and its influence on the denoising performance. Table 5.2 shows the average PSNR on BSD68 [Roth and Black, 2009]

Parameters	Mapping Module Layers					
	1 st	2 nd	3 rd	4 th	5 th	6 th
Padding	1	3	3	3	3	3
Dilation	1	3	3	3	3	3
Kernel Size	3	3	3	3	3	3
Channels	64	64	64	64	64	64

Table 5.1: Detailed architecture of an identity mapping module.

Training patch size					
20	30	40	50	60	70
29.13	29.30	29.34	29.36	29.37	29.38

Table 5.2: Denoising performance (in PSNR) on the BSD68 dataset [Roth and Black, 2009] for different sizes of training input patches for $\sigma_n = 25$, keeping all other parameters constant.

Number of modules				
2	3	4	6	8
29.28	29.34	29.34	29.35	29.36

Table 5.3: The average PSNR accuracy of the denoised images for the BSD68 dataset, with respect to different number of modules \mathbf{M} . The higher the number of modules, the higher is the accuracy.

for $\sigma_n = 25$ with respect to the increase in size of the training patch. It is obvious that there is a marginal improvement in PSNR as the patch size increases. The main reason for this phenomenon is the size of the receptive field, with a larger patch size network learns more contextual information, hence able to predict local details better.

5.3.5.2 Number of Modules

We show the effect of the number of modules on denoising results. As mentioned earlier, each module \mathbf{M} consists of six convolution layers, by increasing the number of modules, we are making our network deeper. In this settings, all parameters are constant, except the number of modules as shown in Table 5.3. It is clear from the results that making the network deeper increase the average PSNR. However, since fast restoration is desired, we prefer a small network of three modules *i.e.* $\mathbf{M} = 3$, which achieves better performance than other methods.

5.3.5.3 Kernel Dilation and Number of Layers

It has been shown that the performance of some networks can be improved either by increasing the depth of the network or by using large convolution filter size to

No of layers	18	9	6
Kernel dilation	1	2	3
	29.34	29.34	29.34

Table 5.4: Denoising performance for different network settings to dissect the relationship between kernel dilation, number of layers and receptive field.

Dilation	Identity	Boosting	PSNR
No	No	No	29.24
Yes	No	No	29.23
No	Yes	No	29.28
Yes	Yes	No	29.32
Yes	Yes	Yes	29.34

Table 5.5: PSNR reported on the BSD68 dataset for $\sigma_n = 25$ when different features are added to the DnCNN baseline (first row).

capture the context information [Zhang et al., 2017b,a]. This helps the restoration of noisy structures in the image. The usage of traditional 3×3 filters is popular in deeper networks. However, using dilated filters there is a tradeoff between the number of layers and the size of the dilated filters without effecting denoising results. In Table 5.4, we present three experimental settings to show the tradeoff between the dilated filter size and the depth of network. In the first experiment as shown in the first column of Table 5.4, we use a traditional filter of size 3×3 and depth of 18 to cover the receptive field of training patch of size 40. In the next experiment, we keep the size of the filter same but enlarge the filter using a dilation factor of two. This increases the size of the filter to 5×5 but having nine non-zero entries it can be interpreted as a sparse filter. Therefore, the receptive field of the training patch can now be covered by nine non-linear mapping layers, contrary to the 18-layers depth per module. Similarly, by expanding the filter by a dilation of three would result in the depth of each module to be six. As in Table 5.4, all three trained models result in similar denoising performance, with the obvious advantage of the shallow network being the most efficient. The number of parameters reduced from 1954k to 663k, similarly, the memory usage for one input patch is reduced to 22MB to 6.5MB.

5.3.5.4 Combination of Kernel Dilation, Identity Connection and Boosting

In Table 5.5, we show the performance on BSD68 dataset when adding different features including a kernel dilation of three across all convolution layers, identity skip connection, or boosting via geometric transformation to the DnCNN baseline which is reported in the first row. The improvement over DnCNN is observed with the introduction of identity skip connections. Applying a dilation of three over 17 or

	Jiao <i>et al.</i>	Bae <i>et al.</i>	DnCNN	Ours
Kernel Size	3x3	3x3	3x3	3x3
Patch Size	40x40	40x40	40x40	40x40
Channels	64	320	64	64
Training data	BSD400	BSD400 +Urban100	BSD400	BSD400
BatchNorm	Yes	Yes	Yes	No
Conv. layers	20	20	17	19
No. parameters	671k	16629k	566k	630k
BSD68	-	26.3 dB	26.2 dB	26.4 dB
BSD100	29.0 dB	-	29.0 dB	29.2 dB

Table 5.6: Comparisons with state-of-the-art methods on BSD68 with $\sigma_n = 50$, and BSD100 with $\sigma_n = 25$. The results of [Bae et al., 2017] and [Jiao et al., 2017] are taken from their respective papers.

19 convolutional layers of DnCNN (row 2) does not appear to be effective. However, using dilated convolution in a short chain of six layers, such as row 3, improves the performance further. In Table 5.5, PSNR is 29.32 dB without boosting and 29.34 dB (last row) if we average the output from eight transformed images.

5.3.6 Comparisons

In this section, first we demonstrate how our method performs on classical images and then report results on the BSD68 dataset.

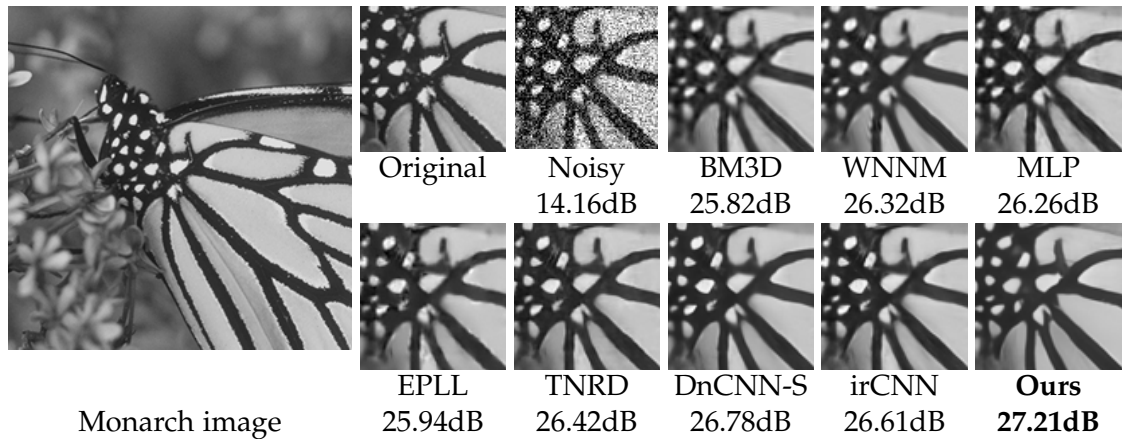


Figure 5.3: Denoising quality comparison on a sample image with strong edges and texture, selected from classical image set for noise level $\sigma_n = 50$. The visual quality, *i.e.* sharpness of the edges on the wings and small textures reproduced by our method is the better than others.

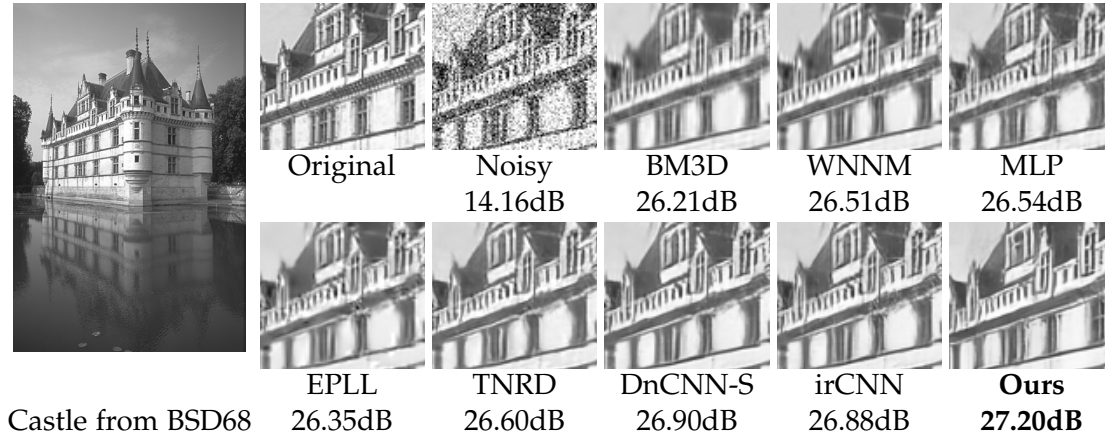


Figure 5.4: Comparison on a sample image from BSD68 dataset [Roth and Black, 2009] for $\sigma_n = 50$. Our network is able to recover fine textures in the background and on the castle, while other methods cannot reproduce such textures accurately.

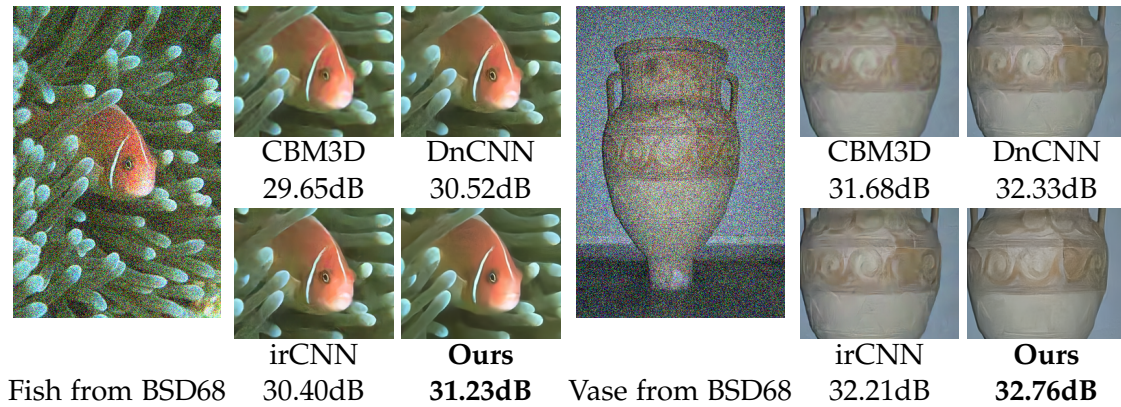


Figure 5.5: Denoising performance for state-of-the-art versus the proposed method on sample color images from the dataset in [Roth and Black, 2009], where the noise standard deviation σ_n is 50. The image we recover is more natural, contains less contrast artifacts and is closest to the ground-truth.

	Cman	House	Peppers	Starfish	Monar	Airpl	Parrot	Lena	Barbara	Boat	Man	Couple	Average
$\sigma_n = 15$													
BM3D	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.372
WNNM	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.696
EPLL	31.85	34.17	32.64	31.13	32.10	31.19	31.42	33.92	31.38	31.93	32.00	31.93	32.138
CSF	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	32.08	31.98	32.318
TNRD	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.502
DnCNNs	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.859
DnCNNB	32.10	34.93	33.15	32.02	32.94	31.56	31.63	34.56	32.09	32.35	32.41	32.41	32.680
IrCNN	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.769
Ours-blind	32.11	35.10	33.28	32.31	33.07	31.58	31.80	34.67	32.48	32.42	32.40	32.50	32.812
Ours	32.61	35.21	33.21	32.35	33.33	31.77	32.01	34.69	32.74	32.44	32.50	32.52	32.950
$\sigma_n = 25$													
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.969
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.257
EPLL	29.26	32.17	30.17	28.51	29.39	28.61	28.95	31.73	28.61	29.74	29.66	29.53	29.692
MLP	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.027
CSF	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.837
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.055
DnCNNs	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.436
DnCNNB	29.94	33.05	30.84	29.34	30.25	29.09	29.35	32.42	29.69	30.20	30.09	30.10	30.362
IrCNN	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.384
Ours-blind	29.87	33.34	30.94	29.68	30.39	29.08	29.38	32.65	30.17	30.27	30.08	30.20	30.505
Ours	30.26	33.44	30.87	29.77	30.62	29.23	29.61	32.66	30.29	30.30	30.18	30.24	30.624
$\sigma_n = 50$													
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.722
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.052
EPLL	26.10	29.12	26.80	25.12	25.94	25.31	25.95	28.68	24.83	26.74	26.79	26.30	26.471
MLP	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.783
TNRD	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.812
DnCNNs	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.178
DnCNNB	27.03	30.02	27.39	25.72	26.83	25.89	26.48	29.38	26.38	27.23	27.23	26.91	27.206
IrCNN	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.136
Ours-blind	27.03	30.48	27.57	26.01	27.03	25.84	26.53	29.77	26.89	27.28	27.29	27.06	27.398
Ours	27.25	30.70	27.54	26.05	27.21	26.06	26.53	29.65	26.62	27.36	27.26	27.24	27.457
$\sigma_n = 70$													
BM3D	24.62	27.91	25.07	23.56	24.24	23.75	24.49	27.57	25.47	25.40	25.56	25.00	25.221
WNNM	24.86	28.59	25.25	23.78	24.62	24.00	24.64	27.85	26.17	25.58	25.68	25.18	25.517
EPLL	24.60	27.32	25.03	23.52	24.19	23.72	24.44	27.11	23.20	25.27	25.50	24.80	24.891
DnCNNs	25.37	28.22	25.50	23.97	25.10	24.34	24.98	27.85	23.97	25.76	25.91	25.31	25.523
Ours	25.83	29.19	25.90	24.28	25.66	24.59	25.12	28.25	25.06	26.00	26.02	25.78	25.974

Table 5.7: Performance comparison between image denoising algorithms on widely used classical images, in terms of PSNR (in dB). The best results are highlighted with bold red color while the blue color represents the second best denoising results.

5.3.6.1 Classical Images

For completeness, we compare our algorithm to several state-of-the-art denoising methods using grayscale classical images shown in Figure 5.3 and reported in Table 5.7.

In Table 5.7, we present the average PSNR for the denoised images. Our network is the best performer for almost all classical images except “Barbara”. The reason for this may be the repetitive structures in the mentioned image, which makes it easy for BM3D [Dabov et al., 2007b] and WNNM [Gu et al., 2014] to find and employ patches

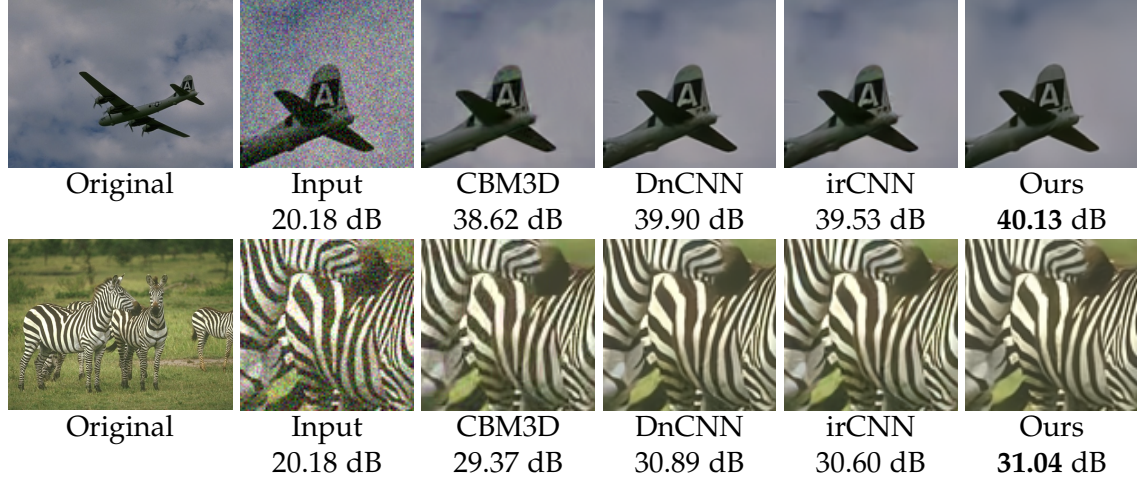


Figure 5.6: A sample color image with rich textures, selected from the BSD68 dataset [Roth and Black, 2009] for $\sigma_n = 25$. On a magnified view, the image our network recovers is sharper than those generated by most of the methods

Noise Levels	Methods								
	BM3D	WNNM	EPLL	TNRD	DnCNNs	IrCNN	NLNet	Ours agnostic	Ours specific
15	31.08	31.32	31.19	31.42	31.73	31.63	31.52	31.68	31.81
25	28.57	28.83	28.68	28.92	29.23	29.15	29.03	29.18	29.34
50	25.62	25.83	25.67	26.01	26.23	26.19	26.07	26.31	26.40
70	24.44	-	24.43	-	24.90	-	-	-	25.13

Table 5.8: Performance comparison between our method and existing algorithms on the grayscale version of the BSD68 dataset [Roth and Black, 2009]. The missing denoising results, indicated by “-”, occurs when the method is not trained to deal with the input noisy images.

with great similarity to the noisy input, hence providing better results.

Subsequently, we depict an example from the classical images. The visual quality of our recovered images, as shown in Figure 5.3, is better than all others. This also illustrates that our network restores aesthetically pleasing textures. Small and noticeable features restored by our network include the sharpness and the clarity of the subtle textures around the fore and hind wings, mouth, and antennas of the butterfly. Furthermore, a magnified view of the results in Figures 5.3 for methods such as [Dabov et al., 2007b; Zhang et al., 2017b; Lefkimmatis, 2016] shows artifacts and failures in the smooth areas. Our CNN network also outperforms [Zhang et al., 2017b; Lefkimmatis, 2016; Zhang et al., 2017a], which are trained using deep neural networks.

Noise Levels	Methods							
	CBM3D	MLP	TNRD	DnCNN	IrCNN	CNLNet	Ours agnostic	Ours specific
15	33.50	-	31.37	33.89	33.86	33.69	33.96	34.12
25	30.69	28.92	28.88	31.33	31.16	30.96	31.32	31.42
50	27.37	26.00	25.94	27.97	27.86	27.64	28.05	28.19

Table 5.9: The similarity between the denoised color images and the ground-truth color images of the BSD68 dataset for our method and existing algorithms measured by PSNR (in dB) reported for noise levels of $\sigma=15$, 25, and 50.

5.3.6.2 BSD68 Dataset

We present the average PSNR scores for the estimated denoised images in Table 5.8. The IRCNN [Zhang et al., 2017b] and DnCNN [Zhang et al., 2017a] network structures are similar, hence produce nearly similar results. On the other hand, our method reconstructs the images accurately, achieving higher PSNR than completing methods on all four levels of noise. Furthermore, the difference in PSNR between our method and the state-of-the-art techniques at the higher noise levels.

For a comprehensive evaluation, we demonstrate the visual results on a selected grayscale image from BSD68 [Roth and Black, 2009] dataset in Figure 5.4. In our results, the image details are more similar to the ground-truth details, and our quantitative results are numerically higher than the others. Our method outperforms the second best method by several orders of magnitude (PSNR is computed in the logarithmic scale). Also, note that the denoising results of other CNN based algorithms are comparable to each other.

5.3.6.3 Color Image Denoising

For noisy color images, we train our network with the noisy RGB input patches of size 40×40 with the corresponding clean ground-truth patches. We only modify the first and last convolution layer of the grayscale network to input and output three channels instead of one channel, keeping all other parameters same as the grayscale network.

We present the quantitative results in Table 5.9 and qualitative results in Figures 5.5 and 5.6 against benchmark methods including the latest CNN based state-of-the-art color image denoising techniques [Zhang et al., 2017a,b; Dabov et al., 2007b]. It can be observed that our algorithm attains an improved average PSNR on all three different noise levels for the color version of BSD68 dataset [Roth and Black, 2009]. As shown, our method restores true colors closer to their authentic values while others fail and induce false colorizations in certain image regions. Furthermore, a close

Metrics	Methods						
	WNNM	EPLL	BM3D	MLP	TNRD	DnCNN	Ours
PSNR	34.44	33.51	34.61	34.14	29.92	32.43	36.04
SSIM	0.8646	0.8244	0.8507	0.8331	0.8306	0.7900	0.9136

Table 5.10: Mean PSNR and SSIM of the denoising methods evaluated on the real images dataset by [Plötz and Roth, 2017].

look reveals that our network reproduces the local texture with much less artifacts and sufficiently sharp details.

5.3.6.4 Darmstadt Noise Dataset: Real-world Images

So far, state-of-the-art denoising methods, such as FormResNet [Jiao et al., 2017], DnCNN [Zhang et al., 2017a], IrCNN [Zhang et al., 2017b] and BM3D [Dabov et al., 2007b] *etc.* have normally been evaluated on classical images and the BSD68 dataset. Recently, [Plötz and Roth, 2017] proposed the Darmstadt Noise Dataset (DND) benchmark for denoising algorithms which consists of 50 images. The dataset is composed of images with interesting and challenging structures. The images are converted to sRGB and gamma correction is applied. The size of each image is in Megapixels; therefore, each image is cropped at 20 locations and each composed of 512×512 pixels yielding 1000 test crops, and overlap between the images is about 10%. Only these test images are provided, there are no images for either training or validation. Therefore, we use the same model which is trained on the synthetic BSD68 [Roth and Black, 2009] dataset. The quantitative results in PSNR and SSIM averaged over all the images for real-world DnD is presented in Table 5.10. It can be observed that our method is the best performer followed by BM3D. Previously, the classical method BM3D is considered to be outperformed by most of the state-of-the-art algorithms on synthetic datasets; however, this is not the case when using the real-world Darmstadt Noise Dataset. It is to be noted that our method does not require to know the noise level in advance unlike BM3D and other state-of-the-art techniques. Furthermore, we visually compare our method with a few recent algorithms as shown on several samples from [Plötz and Roth, 2017] in Figure 5.7¹. It can be observed that both CBM3D [Dabov et al., 2007b], as well as DnCNN [Zhang et al., 2017a], are unable to remove the noise from the images. On the other hand, it can be seen that our method eliminates the noise and preserve the structures.

As a last experiment, we demonstrate the performance of our network on real-world noisy images from [Zhang et al., 2017a]². Figure 5.8 shows such examples

¹PSNR for individual images are not available as [Plötz and Roth, 2017]’s system only provide average PSNR

²PSNR for these images not presented as ground-truth is not available

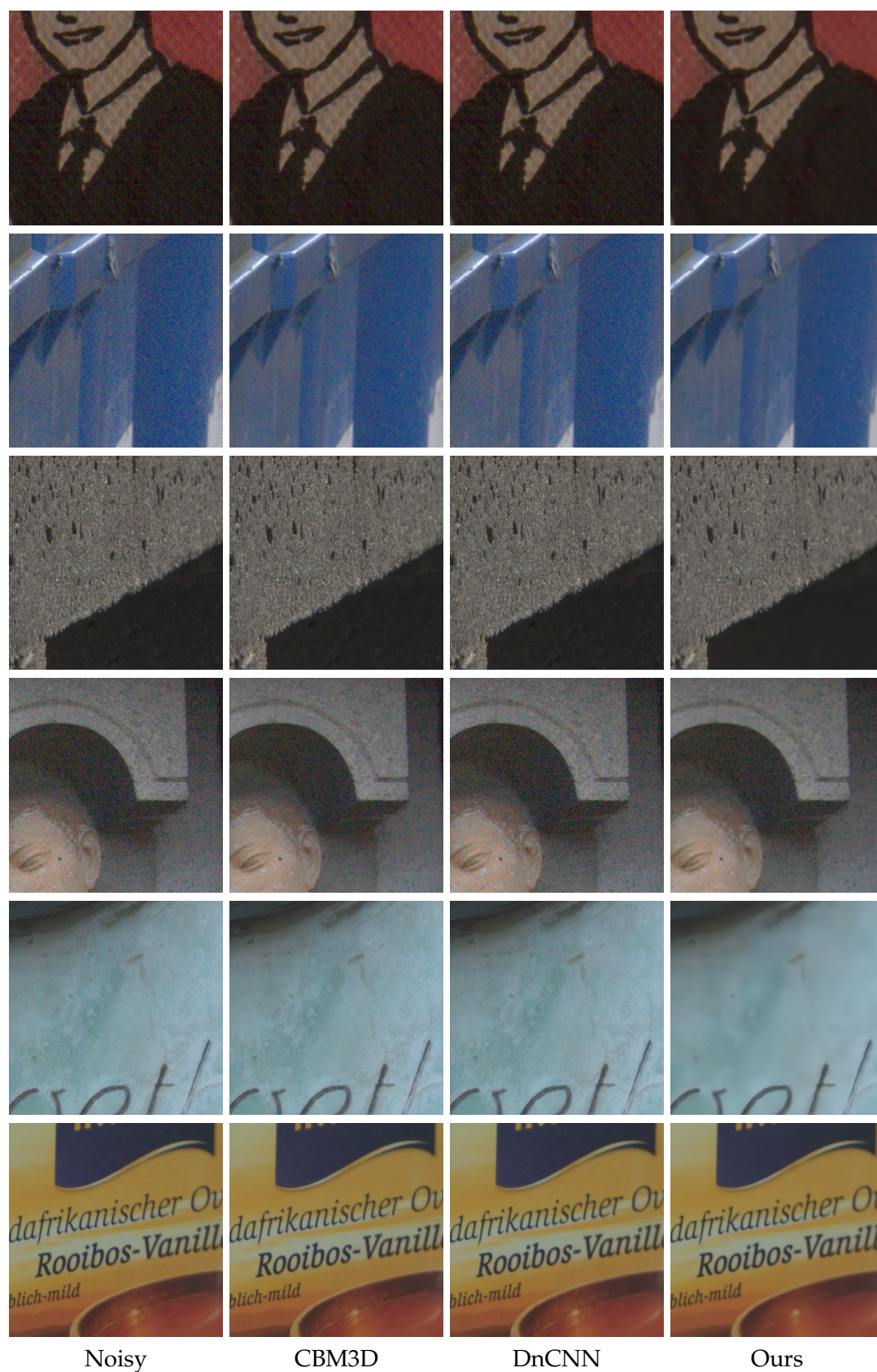


Figure 5.7: Real images from Darmstadt Noise Dataset (DND) benchmark for different denoising algorithms [Plötz and Roth, 2017].

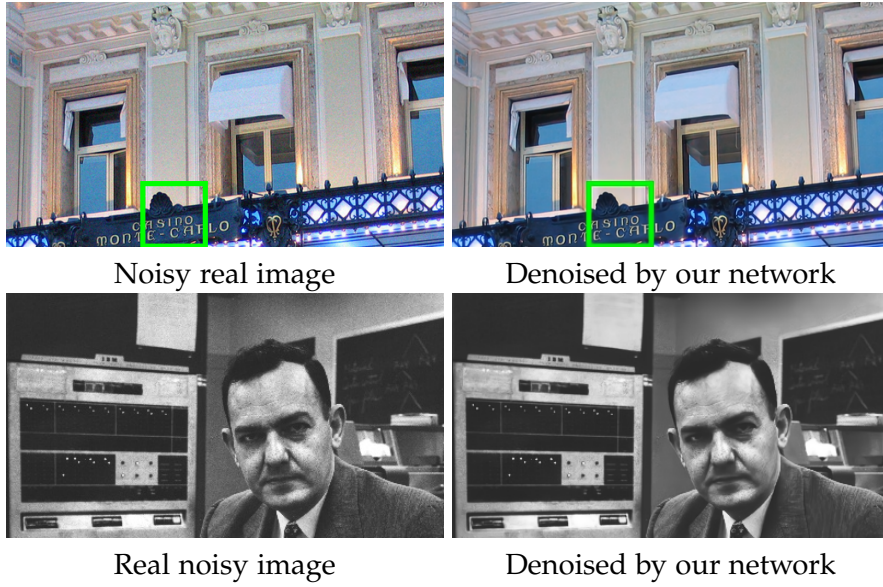


Figure 5.8: Two real images from [Zhang et al., 2017a] denoised by our noise level agnostic color and grayscale models, respectively.

denoised by our noise-level agnostic (requiring no noise prior) denoising models. As visible, the details are preserved properly and the noise is removed effectively. When the noise is Additive white Gaussian noise (AWGN) or adequately satisfies the criteria for additive Gaussian-like noise criteria, our model works accurately as our models are trained on Gaussian noise. This experiment indicates that our network is well-suited for real-world applications.

5.4 Discussion

To sum up, we employ residual learning and identity mapping for image denoising using a deep network consisting of three identity mapping modules each with six ReLU and conv pairs of 19 weight layers with dilated convolutional filters without batch normalization. Our choice of network is based on the ablative studies performed in the experimental section of this chapter.

This is the first modular framework to predict the denoised output without any dependency on the pre or post-processing. Our proposed network removes the potentially authentic image structures while allowing the noisy observations to go through its layers, and learn neural network parameters to predict the noise-free images.

In experiments on synthetic images, we have provided ample examples and have

shown that our network outperforms classical state-of-the-art denoising algorithms that are intended for use on natural images. Furthermore, we have compared against the current convolutional neural networks both visually and numerically. Our network gain is about **0.1dB** on BSD68 dataset compared to the second best performing method and results are visually pleasing.

On real images of the Darmstadt Noise Dataset (DND), we have shown that our method provides visually pleasing results and gains about **1.43dB** of PSNR compared to BM3D (second best performing method). The real images appear less grainy after passing through our proposed network and preserving fine image structures as compared to BM3D and DnCNN. Furthermore, competitive denoising algorithms known noise level in the image. On the contrary, our network does not make an assumption about the specific noise level present in the images.

Conclusion and Future Work

All human wisdom is summed up in two words: wait and hope.

Alexandre Dumas

In this exposition, we have investigated two main image restoration problems: image deblurring and image denoising. To approach these problems we learn novel priors from external images for image enhancement. Our methods excel and prove useful for recovering image details for both problems. Our contributions to class-specific deblurring and category-specific denoising, heavily rely on class-specific datasets while our CNN denoiser makes use of large generic natural image datasets. Regarding image quality, our methods can furnish a reliable outcome compared to state-of-the-art techniques, which are conservative by design. Consequently, the various applications in this thesis complement conventional methods and push the envelope and formulate new approaches for contemporary research in the field. In the next sections, we conclude this exposition and list our contributions as well as present potential future directions.

6.1 Conclusion

As mentioned beforehand in this dissertation, we have introduced three new efficient and effective approaches for image enhancement compared to the state-of-the-art methods. We summarize the contributions for each chapter in the following paragraphs.

Chapter 3: Class-Specific image deblurring In this chapter, we introduced our novel image deblurring algorithm to recover attenuated frequencies using class-specific external datasets. The purpose of image deblurring algorithms is to reliably

recover distinct spatial frequencies suppressed by the blurring kernel. Existing algorithms relies on salient image features, for example, gradients and edges. Throughout this work, we devise a class-specific prior based on the band-pass filter responses and incorporate it into a deblurring strategy. More specifically, we show that the subspace of band-pass filtered images and their intensity distributions serve as useful priors for recovering image frequencies that are difficult to recover by generic image priors.

We presented an insight into our algorithm and discussed different aspects and its implication on our method. We have also shown the effect of the blur kernel on frequencies present in the image. We have provided the convergence and drove the complexity of our algorithm as well as provided the running time. Furthermore, we also present the deblurring results with and without our prior which proves the effectiveness of our framework.

Finally, we illustrated that our image deblurring framework, when equipped with the introduced priors, significantly outperforms many state-of-the-art methods which are using generic image priors or class-specific exemplars. We also provide few examples of real-world images. We, therefore, believe that our class-specific image deblurring can be used to deblur objects of interest rather than deblurring the whole scene or image.

Chapter 4: Category-specific image denoising In this chapter, we presented a novel image denoising algorithm that uses external, category specific image database. Current image restoration algorithms search patches either from a generic database or noisy image itself. Our method first selects clean images similar to the noisy image from a database of the same class. Then, it assembles a set of “support patches” from the selected images within the spatial locality of each noisy patch. These clean and noise-free support patches resemble the noisy patch and correspond principally to the identical part of the depicted object.

Additionally, we employed a content adaptive distribution model for each patch where we derive the parameters of the distribution from the support patches. We formulate noise removal task as an optimization problem in the transform domain. Our objective function composed of three terms: data fidelity that imposes similarity between the noisy patch and the clean patch, a Gaussian fidelity term that imposes category specific information, and a low-rank term that encourages the similarity between the noisy and the support patches in a robust manner.

Furthermore, we analyze the influence of the number and the size of the support patches on our algorithm. Similarly, we examine the relative importance of Gaussian fidelity term and low-rank term in the experimental section. Moreover, we illustrate

the role of external datasets and the effect of external dataset size on denoising performance. Afterward, we check the sensitivity of our algorithm against the variation in the pose of the object in the given external dataset.

Finally, we show experiments on five different object categories. The performance confirms the advantage of incorporating category-specific information in noise removal. Our method demonstrates the superior performance against state-of-the-art alternatives on grayscale and color images.

Chapter 5: CIMM for image denoising In this chapter, we introduced a fully-convolutional network model consists of a Chain of Identity Mapping Modules (CIMM) for image denoising. The CIMM have two distinctive properties which play an essential part in removing noise from images. Firstly, each residual unit, also known as mapping module, employs skip connections and secondly, the dilated kernels utilized in convolutional kernels. We also used pre-activation as compared to generally applied post-activation to preserve the gradient magnitude propagated in both the forward and backward directions.

We analyzed different aspects of our CNN model. We showed that the influence of increasing and/or decreasing the input patch size, number of modules, and number of layers. We also provided the relation between the kernel dilation and the number of layers in a mapping module. Due to kernel dilation, the depth of the network decreased, capturing the context information making the network more efficient.

Finally, the proposed network produces remarkably higher numerical accuracy and better visual image quality against the state-of-the-art classical and CNN algorithms after being trained on the BSD400 dataset [Martin et al., 2001]. Similarly, when the proposed algorithm is evaluated numerically on classical images, the BSD68 dataset [Martin et al., 2001] and real-world images from Darmstadt Noise Dataset (DND) [Plötz and Roth, 2017] giving state of the art results.

6.2 Future Directions

We consider several future directions in which to extend the work presented in this dissertation.

Class-Specific image deblurring

- At the current state, our class-Specific image deblurring algorithm focuses on deblurring of images containing a single object using a class-specific training dataset. In the future, this work can be extended to deal with multiple objects.

This could be achieved by first localizing and classifying the different objects in the image, and deblurring each object region separately using the training data for the corresponding class. Furthermore, it is worth investigating whether, and if so, to what extent, class-specific training data is required as opposed to generic training data.

- Our algorithm is currently limited by the assumption of a spatially uniform blur. In the future, we would like to extend our blur model to handle non-uniform blur caused by camera motion, rotation and defocus. This extension requires the geometrical and physical modeling of image formation in the above circumstances.
- Class-specific blind deblurring can handle outliers (such as pixel saturation, non-Gaussian noise, dead pixels, hot pixel *etc.*) in images in a limited capacity. Outliers have a significant influence on kernel estimation. Therefore, in future work, it is essential to incorporate outlier handling in our algorithm for kernel estimation step to achieve robustness against various outliers.
- Our algorithm currently know the object class before deblurring, it will be worth investigating to deblur the object without knowing the object class beforehand.
- It will be worth investigating whether Convolutional Neural Networks (CNN) and Gaussian Adversarial Network (GANs) play any role in removing the blur from the images.

Category-specific image denoising

- An important question that requires more discussion is the behavior and sensitivity of the algorithm to significant variations in pose, facial expressions, size, illumination, and view angle. Seeking an answer to this question will help us in improving the robustness of the algorithm. This aspect of our method will be studied in our future work.
- External category-specific datasets are essential for the success of the proposed method. Commonly, the datasets are available; however, if due to some reason it is difficult to find one, then image retrieval algorithms can be used to achieve the desired outcome.
- Another exciting aspect to study is the combination of category-specific external image denoising and the CNN based internal image denoising.

-
- Another worth exploring aspect of our approach is the distance metric used in finding similar patches in the external dataset. Currently, we employ the Euclidean distance *i.e.* ℓ_2 -norm for searching similarity between the noisy patch and other suitable candidates. Though, this metric has a disadvantage which is that low value may not reflect the similarity between the candidate patch and the noisy patch [Domingos, 2012]. Therefore incorporation of metric learning [McFee and Lanckriet, 2010] and deep learning [Chopra et al., 2005; Reed et al., 2015] may improve the performance by finding the similar patches accurately. The idea is that the returned clean patches must have similar rankings to the noisy patch. Similarly, in case of deep learning, a mapping can be done using only similar noisy patches which can then be extended to clean versions.
 - Lastly, we can also explore patch search solutions to improve the efficiency of the algorithm.

CIMM for image denoising

- In the future, we aim to utilize our proposed CNN to other image restoration and enhancement tasks such as deblurring, color correction, JPEG artifact removal, rain removal, dehazing and super-resolution *etc.*
- At present, our approach is only applicable to Gaussian noise removal. However, we would like to train our model with different noise types such as Poisson noise, astronomical noise *etc.* and examine its performance on these specific noise types. It should be noted that other state-of-the-art methods, for example, BM3D [Dabov et al., 2007b], WNNM [Gu et al., 2014] are only applicable to Gaussian noise and may not be readily adapted to handle different noise types.
- CNN approaches perform ineffectively on images with regular and repeating structures when compared against classical denoising methods. One such example is “Barbara” image, where CNN algorithms are lacking in terms of PSNR. This phenomenon is due to the design of traditional denoising methods to exploit the regular and repeating structures. To overcome this issue, either a block-matching scheme can be incorporated into our CNN approach or relying on consolidating the outcome of various denoising algorithms with our CNN approach.
- Another direction worth exploring is integrating local patch similarity heuristics into CNN.
- In future, we also aim to utilize Gaussian Adversarial Network (GANs) to recover the original image from a noisy one.

Bibliography

- ANSCOMBE, F. J., The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35, 3/4 (1948), 246–254. (cited on page 11)
- ANWAR, S.; HUYNH, C. P.; AND PORIKLI, F., Class-specific image deblurring. *International Conference on Computer Vision*, (2015), 495–503. (cited on pages 13 and 35)
- ANWAR, S.; HUYNH, C. P.; AND PORIKLI, F., Chaining Identity Mapping Modules for Image Denoising. *ArXiv e-prints*, (Dec. 2017). (cited on page 13)
- ANWAR, S.; HUYNH, C. P.; AND PORIKLI, F., Combined Internal and External Category-Specific Image Denoising. *British Machine Vision Conference*, (2017). (cited on page 13)
- ANWAR, S.; HUYNH, C. P.; AND PORIKLI, F., Image Deblurring with a Class-Specific Prior. *IEEE transactions on pattern analysis and machine intelligence*, (2018). (cited on page 13)
- ANWAR, S.; PORIKLI, F.; AND HUYNH, C. P., Category-Specific Object Image Denoising. *Transactions on Image Processing*, 26, 11 (2017), 5506–5518. (cited on pages 13, 108, and 109)
- ARIAS-CASTRO, E.; DONOHO, D. L.; ET AL., Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, (2009). (cited on pages 29 and 30)
- AYERS, G. AND DAINTY, J. C., Iterative blind deconvolution method and its applications. *Optics letters*, 13, 7 (1988), 547–549. (cited on pages 3 and 20)
- BAE, W.; YOO, J.; AND YE, J. C., Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Computer Vision and Pattern Recognition Workshops*, 145–153. (cited on pages xxiv, 41, and 117)
- BAKER, S. AND KANADE, T., Limits on super-resolution and how to break them. *Transactions on Pattern Analysis and Machine Intelligence*, 24, 9 (2002), 1167–1183. (cited on page 5)

- BENGIO, Y.; SIMARD, P.; AND FRASCONI, P., Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks*, (1994). (cited on page 108)
- BERTALMIO, M.; SAPIRO, G.; CASELLES, V.; AND BALLESTER, C., Image inpainting. In *Conference on Computer graphics and Interactive Techniques*, 417–424. (cited on page 2)
- BUADES, A.; COLL, B.; AND MOREL, J.-M., A Non-Local Algorithm for Image Denoising. In *Computer Vision and Pattern Recognition*, 60–65. (cited on pages 11, 28, 31, 36, 86, 94, 107, and 111)
- BURGER, H. C., 2013. *Modelling and learning approaches to image denoising*. Ph.D. thesis, Universität Tübingen Tübingen. (cited on page 29)
- BURGER, H. C.; SCHULER, C. J.; AND HARMELING, S., Image denoising: Can plain Neural Networks compete with BM3D? In *Computer Vision and Pattern Recognition*, 2392–2399. (cited on pages 39, 108, and 113)
- CAI, J.-F.; CANDÈS, E. J.; AND SHEN, Z., A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM J. on Optimization*, (2010), 1956–1982. (cited on page 91)
- CAI, J.-F.; JI, H.; LIU, C.; AND SHEN, Z., Blind motion deblurring from a single image using sparse approximation. In *Computer Vision and Pattern Recognition*, 104–111. (cited on page 69)
- CAI, J.-F.; JI, H.; LIU, C.; AND SHEN, Z., Framelet-based blind motion deblurring from a single image. *Transactions on Image Processing*, 21, 2 (2012), 562–572. (cited on pages 68, 69, 70, and 76)
- CANDÈS, E. AND RECHT, B., Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, (2009), 717–772. (cited on page 89)
- CANNY, J., A computational approach to edge detection. In *Readings in Computer Vision*, 184–203. Elsevier. (cited on page 3)
- CASELLES, V.; SAPIRO, G.; AND CHUNG, D. H., Vector median filters, inf-sup operations, and coupled PDE’s: Theoretical connections. *Journal of Mathematical Imaging and Vision*, 12, 2 (2000), 109–119. (cited on page 29)
- CHAKRABARTI, A., A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, 221–235. (cited on pages xvii and 27)

-
- CHAN, S. H.; ZICKLER, T.; AND LU, Y. M., Monte Carlo non-local means: Random sampling for large-scale image filtering. *Transaction on Image Processing*, (2014), 3711–3725. (cited on page 34)
- CHAN, T. F. AND WONG, C.-K., Total variation blind deconvolution. *Transactions on Image Processing*, 7, 3 (1998), 370–375. (cited on pages 18 and 20)
- CHANG, S. G.; YU, B.; AND VETTERLI, M., Adaptive wavelet thresholding for image denoising and compression. *Transactions on image processing*, 9, 9 (2000), 1532–1546. (cited on page 37)
- CHATTERJEE, P. AND MILANFAR, P., Is Denoising Dead? *Transactions on Image Processing*, (2010), 895–911. (cited on page 28)
- CHATTERJEE, P. AND MILANFAR, P., Patch-based near-optimal image denoising. *Transactions on Image Processing*, 21, 4 (2012), 1635–1649. (cited on page 11)
- CHEN, F.; ZHANG, L.; AND YU, H., External Patch Prior Guided Internal Clustering for Image Denoising. (2015). (cited on pages 36 and 38)
- CHEN, S.; BILLINGS, S. A.; AND LUO, W., Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50, 5 (1989), 1873–1896. (cited on page 36)
- CHEN, Y. AND POCK, T., Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *Transactions on Pattern Analysis and Machine Intelligence*, 39, 6 (2017), 1256–1272. (cited on pages 39 and 113)
- CHO, S. AND LEE, S., Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, vol. 28, 145. (cited on pages 20, 44, 49, 57, 68, 69, 70, and 76)
- CHO, T. S.; PARIS, S.; HORN, B. K.; AND FREEMAN, W. T., Blur kernel estimation using the radon transform. In *Computer Vision and Pattern Recognition*, 241–248. (cited on page 21)
- CHOPRA, S.; HADSELL, R.; AND LECUN, Y., Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition*, vol. 1, 539–546. (cited on page 131)
- COTTER, S. F.; RAO, B. D.; ENGAN, K.; AND KREUTZ-DELGADO, K., Sparse solutions to linear inverse problems with multiple measurement vectors. *Transactions on Signal Processing*, 53, 7 (2005), 2477–2488. (cited on page 37)

- CRIMINISI, A.; PEREZ, P.; AND TOYAMA, K., Region filling and object removal by exemplar-based image inpainting. *Transactions on Image Processing*, (2004), 1200–1212. (cited on page 2)
- DABOV, K.; FOI, A.; KATKOVNIK, V.; AND EGIAZARIAN, K., Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *International Conference on Image Processing*, vol. 1, I–313. (cited on pages 104 and 109)
- DABOV, K.; FOI, A.; KATKOVNIK, V.; AND EGIAZARIAN, K., Image denoising by sparse 3-D transform-domain collaborative filtering. *Transactions on Image Processing*, (2007), 2080–2095. (cited on pages 11, 28, 31, 36, 86, 94, 107, 111, 113, 119, 120, 121, 122, and 131)
- DABOV, K.; FOI, A.; KATKOVNIK, V.; AND EGIAZARIAN, K., BM3D image denoising with shape-adaptive principal component analysis. In *Signal Processing with Adaptive Sparse Structured Representations*. (cited on pages 28, 31, 94, and 107)
- DALAL, N. AND TRIGGS, B., Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition*, vol. 2, 886–893. (cited on pages xviii, xix, xxiii, 56, 61, 62, 63, 70, and 71)
- DELEDALLE, C.-A.; SALMON, J.; DALALYAN, A. S.; AND CHAMPS-SUR MARNE, F., Image denoising with patch based PCA: local versus global. In *British Machine Vision Conference*, 1–10. (cited on pages 28 and 33)
- DOMINGOS, P., A few useful things to know about machine learning. *Communications of the ACM*, 55, 10 (2012), 78–87. (cited on page 131)
- DONG, W.; LI, X.; ZHANG, D.; AND SHI, G., Sparsity-based image denoising via dictionary learning and structural clustering. In *Computer Vision and Pattern Recognition*, 457–464. (cited on page 36)
- DONG, W.; SHI, G.; AND LI, X., Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *Transactions on Image Processing*, (2013), 700–711. (cited on pages 31, 37, 89, and 92)
- EDELSBRUNNER, H. AND HARER, J., Persistent homology-a survey. *Contemporary mathematics*, 453 (2008), 257–282. (cited on page 41)
- EFROS, A. AND LEUNG, T., Texture synthesis by non-parametric sampling. In *Computer Vision*, 1033–1038. (cited on page 31)

-
- ELAD, M. AND AHARON, M., Image denoising via sparse and redundant representations over learned dictionaries. *Transactions on Image Processing*, (2006), 3736–3745. (cited on pages 28, 35, and 36)
- F. CHEN, L. Z. AND YU, H., External Patch Prior Guided Internal Clustering for Image Denoising. In *International Conference on Computer Vision*, 1211–1218. (cited on pages 35, 38, and 94)
- FERGUS, R.; SINGH, B.; HERTZMANN, A.; ROWEIS, S. T.; AND FREEMAN, W. T., Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, vol. 25, 787–794. (cited on pages 23, 24, 44, 49, 57, 68, 69, 70, 71, and 76)
- FERRARI, V.; JURIE, F.; AND SCHMID, C., From Images to Shape Models for Object Detection. *International Journal on Computer Vision*, 87, 3 (2010), 284–303. (cited on pages xix, 56, 72, 73, 76, and 79)
- FISH, D.; WALKER, J.; BRINICOMBE, A.; AND PIKE, E., Blind deconvolution by means of the Richardson–Lucy algorithm. *JOSA A*, 12, 1 (1995), 58–65. (cited on pages 3 and 20)
- FOI, A.; KATKOVNIK, V.; AND EGIAZARIAN, K., Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *Transactions on Image Processing*, (2007), 1395–1411. (cited on pages 28, 31, and 92)
- FREEMAN, W. T.; JONES, T. R.; AND PASZTOR, E. C., Example-based super-resolution. *Computer graphics and Applications*, 22, 2 (2002), 56–65. (cited on page 3)
- GEMAN, D. AND YANG, C., Nonlinear image recovery with half-quadratic regularization. *Transactions on Image Processing*, 4, 7 (1995), 932–946. (cited on page 19)
- GEORGHIADES, A.; BELHUMEUR, P.; AND KRIEGMAN, D., From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. *Pattern Analysis and Machine Intelligence*, 23, 6 (2001), 643–660. (cited on pages xviii, xix, 56, 62, 74, and 76)
- GEUSEBROEK, J.-M. AND SMEULDERS, A. W. M., A Six-Stimulus Theory for Stochastic Texture. *International Journal on Computer Vision*, 62, 1-2 (2005), 7–16. (cited on page 46)
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; AND MALIK, J., Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 580–587. (cited on page 84)

- GLASBEY, C. A. AND MARDIA, K. V., A review of image-warping methods. *Journal of applied statistics*, 25, 2 (1998), 155–171. (cited on page 2)
- GLASNER, D.; BAGON, S.; AND IRANI, M., Super-Resolution from a Single Image. In *International Conference on Computer Vision*. (cited on page 31)
- GONZALEZ, R. C. AND WOODS, R. E., 1992. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., 2nd edn. ISBN 0201508036. (cited on page 49)
- GOOSSENS, B.; LUONG, H.; PIZURICA, A.; AND PHILIPS, W., An improved non-local denoising algorithm. In *Local and Non-Local Approximation in Image Processing, International Workshop, Proceedings*, 143. (cited on pages 31 and 86)
- GU, S.; ZHANG, L.; ZUO, W.; AND FENG, X., Weighted Nuclear Norm Minimization with Application to Image Denoising. In *Computer Vision and Pattern Recognition*, 2862–2869. (cited on pages 31, 33, 94, 107, 113, 119, and 131)
- HACOHEN, Y.; SHECHTMAN, E.; AND LISCHINSKI, D., Deblurring by Example Using Dense Correspondence. In *International Conference on Computer Vision*, 2384–2391. (cited on pages xx, 19, 26, 77, 78, and 79)
- HARALICK, R. M. AND SHAPIRO, L. G., Image segmentation techniques. *Computer vision, graphics, and image processing*, 29, 1 (1985), 100–132. (cited on page 3)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., Deep Residual Learning for Image Recognition. CoRR, abs/1512.03385 (2015). (cited on page 110)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. CoRR, abs/1502.01852 (2015). (cited on page 114)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 770–778. (cited on page 108)
- HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016b. *Identity Mappings in Deep Residual Networks*, 630–645. (cited on pages 110 and 111)
- HEWITT, E. AND HEWITT, R. E., The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis. *Archive for history of Exact Sciences*, 21, 2 (1979), 129–160. (cited on page 44)
- HIRSCHMÜLLER, H. AND SCHARSTEIN, D., Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition*, 1–8. (cited on pages xx, xxi, 94, 100, 101, and 102)

-
- HUNT, B. R., The application of constrained least squares estimation to image restoration by digital computer. *Transactions on Computers*, 100, 9 (1973), 805–812. (cited on page 16)
- JIAO, J.; TU, W.-C.; HE, S.; AND LAU, R. W., Formresnet: Formatted residual learning for image restoration. In *Computer Vision and Pattern Recognition Workshops*, 1034–1042. (cited on pages xxiv, 41, 117, and 122)
- JOSHI, N.; MATUSIK, W.; ADELSON, E. H.; AND KRIEGMAN, D. J., Personal photo enhancement using example images. *ACM Trans. Graph*, 29, 2 (2010), 12. (cited on page 26)
- JOSHI, N.; SZELISKI, R.; AND KRIEGMAN, D., PSF estimation using sharp edge prediction. In *Computer Vision and Pattern Recognition*, 1–8. (cited on page 21)
- JOSHI, N.; ZITNICK, C. L.; SZELISKI, R.; AND KRIEGMAN, D. J., Image deblurring and denoising using color priors. In *Computer Vision and Pattern Recognition*, 1550–1557. (cited on page 5)
- KIM, J.; KWON LEE, J.; AND MU LEE, K., Accurate image super-resolution using very deep convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, 1646–1654. (cited on page 3)
- KIM, J.; KWON LEE, J.; AND MU LEE, K., Accurate image super-resolution using very deep convolutional networks. In *Computer Vision and Pattern Recognition*. (cited on page 111)
- KIM, S.-J.; KOH, K.; LUSTIG, M.; BOYD, S.; AND GORINEVSKY, D., An Interior-Point Method for Large-Scale L1-Regularized Least Squares. *Journal of Selected Topics in Signal Processing*, 1, 4 (2007), 606–617. (cited on pages 50, 53, and 54)
- KINGMA, D. P. AND BA, J., Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980 (2014). (cited on page 114)
- KNAUS, C. AND ZWICKER, M., Dual-domain image denoising. In *International Conference on Image Processing*, 440–444. (cited on pages 28 and 34)
- KNAUS, C. AND ZWICKER, M., Progressive Image Denoising. *Transactions on Image Processing*, (2014), 3114–3125. (cited on pages 28 and 34)
- KRAUSE, J.; STARK, M.; DENG, J.; AND FEI-FEI, L., 3D Object Representations for Fine-Grained Categorization. In *International Conference on Computer Vision Workshop*, 554–561. (cited on pages xix, 56, 69, 71, and 94)

- KRISHNAN, D. AND FERGUS, R., Fast Image Deconvolution using Hyper-Laplacian Priors. In *Neural Information Processing Systems*, 1033–1041. (cited on pages 19, 21, 44, 57, and 66)
- KRISHNAN, D.; TAY, T.; AND FERGUS, R., Blind deconvolution using a normalized sparsity measure. In *Computer Vision and Pattern Recognition*, 233–240. (cited on pages 22, 44, 57, 68, 69, 70, 71, 75, and 76)
- KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 1097–1105. (cited on page 84)
- LEBRUN, M.; BUADES, A.; AND MOREL, J.-M., A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, (2013), 1665–1688. (cited on pages 28, 31, 33, 86, and 104)
- LEBRUN, M.; COLOM, M.; BUADES, A.; AND MOREL, J.-M., Secrets of image denoising cuisine. *Acta Numerica*, 21 (2012), 475–576. (cited on page 15)
- LEE, J.-S., Digital image enhancement and noise filtering by use of local statistics. *Transactions on Pattern Analysis and Machine Intelligence*, , 2 (1980), 165–168. (cited on pages 3 and 30)
- LEFKIMMIATIS, S., Non-local color image denoising with convolutional neural networks. *Computer Vision and Pattern Recognition*, (2016). (cited on pages 11, 39, 41, 108, 109, 111, 113, and 120)
- LEVIN, A., Blind motion deblurring using image statistics. In *Neural Information Processing Systems*, 841–848. (cited on page 44)
- LEVIN, A., Blind Motion Deblurring Using Image Statistics. In *Neural Information Processing Systems*, 841–848. (cited on page 46)
- LEVIN, A.; FERGUS, R.; DURAND, F.; AND FREEMAN, W. T., Image and Depth from a Conventional Camera with a Coded Aperture. *ACM Trans. Graph.*, 26, 3 (2007). (cited on pages 18, 19, 26, 44, and 57)
- LEVIN, A. AND NADLER, B., Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition*, 2833–2840. (cited on pages 11, 28, and 34)
- LEVIN, A.; NADLER, B.; DURAND, F.; AND FREEMAN, W. T., Patch Complexity, Finite Pixel Correlations and Optimal Denoising. In *European Conference on Computer Vision*, 73–86. (cited on pages 28 and 34)

-
- LEVIN, A.; WEISS, Y.; DURAND, F.; AND FREEMAN, W. T., Understanding and evaluating blind deconvolution algorithms. In *Computer Vision and Pattern Recognition*, 1964–1971. (cited on page 56)
- LEVIN, A.; WEISS, Y.; DURAND, F.; AND FREEMAN, W. T., Efficient marginal likelihood optimization in blind deconvolution. In *Computer Vision and Pattern Recognition*, 2657–2664. (cited on pages 44, 53, 57, 68, 69, 70, 71, 76, and 77)
- LEVIN, A.; WEISS, Y.; DURAND, F.; AND FREEMAN, W. T., Understanding Blind Deconvolution Algorithms. *Pattern Analysis and Machine Intelligence*, 33, 12 (2011), 2354–2367. (cited on pages 24, 44, and 66)
- LIM, B.; SON, S.; KIM, H.; NAH, S.; AND LEE, K. M., Enhanced deep residual networks for single image super-resolution. In *Computer Vision and Pattern Recognition Workshops*. (cited on page 108)
- LIN, T.; DOLLÁR, P.; GIRSHICK, R. B.; HE, K.; HARIHARAN, B.; AND BELONGIE, S. J., Feature Pyramid Networks for Object Detection. *CoRR*, abs/1612.03144 (2016). (cited on page 110)
- LOU, Y.; FAVARO, P.; SOATTO, S.; AND BERTOZZI, A., Nonlocal similarity image filtering. In *International Conference on Image Analysis and Processing*, 62–71. Springer. (cited on page 31)
- LOWE, D. G., Distinctive Image Features from Scale-Invariant Keypoints. *International Journal on Computer Vision*, 60, 2 (2004), 91–110. (cited on page 35)
- LUO, E.; CHAN, S. H.; AND NGUYEN, T. Q., Image denoising by targeted external databases. In *International Conference on Acoustics, Speech and Signal Processing*, 2450–2454. (cited on page 35)
- LUO, E.; CHAN, S. H.; AND NGUYEN, T. Q., Adaptive Image Denoising by Targeted Databases. *Transactions on Image Processing*, (2015), 2167–2181. (cited on pages 35, 84, 94, 95, 100, and 108)
- LUO, E.; CHAN, S. H.; AND NGUYEN, T. Q., Adaptive Image Denoising by Mixture Adaptation. *Transaction on Image Processing*, 25, 10 (Oct 2016). (cited on page 38)
- MAHMOUDI, M. AND SAPIRO, G., Fast image and video denoising via nonlocal means of similar neighborhoods. *Signal Processing Letters*, (2005), 839–842. (cited on page 97)

- MAIRAL, J.; BACH, F.; PONCE, J.; SAPIRO, G.; AND ZISSERMAN, A., Non-local sparse models for image restoration. In *International Conference on Computer Vision*, 2272–2279. (cited on page 36)
- MARTIN, D.; FOWLKES, C.; TAL, D.; AND MALIK, J., A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *International Conference on Computer Vision*, 416–423. (cited on pages 107, 113, and 129)
- McFEE, B. AND LANCKRIET, G. R., Metric learning to rank. In *International Conference on Machine Learning*, 775–782. (cited on page 131)
- MICHAELI, T. AND IRANI, M., Blind Deblurring Using Internal Patch Recurrence. In *European Conference on Computer Vision*, 783–798. (cited on pages 24 and 25)
- MILANFAR, P., A tour of modern image filtering: New insights and methods, both practical and theoretical. *Signal Processing Magazine*, 30, 1 (2013), 106–128. (cited on page 15)
- MILLER, K., Least squares methods for ill-posed problems with a prescribed bound. *SIAM Journal on Mathematical Analysis*, 1, 1 (1970), 52–74. (cited on page 16)
- MOSLEH, A.; LANGLOIS, J. P.; AND GREEN, P., Image Deconvolution Ringing Artifact Detection and Removal via PSF Frequency Analysis. In *European Conference on Computer Vision*, vol. 8692, 247–262. (cited on page 44)
- MOSSERI, I.; ZONTAK, M.; AND IRANI, M., Combining the power of internal and external denoising. In *International Conference on Computational Photography*, 1–9. (cited on pages 35 and 36)
- MUJA, M. AND LOWE, D. G., Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence*, (2014), 2227–2240. (cited on page 97)
- MURESAN, D. D. AND PARKS, T. W., Adaptive principal components and image denoising. In *International Conference on Image Processing*, 101–104. (cited on page 33)
- NISHIYAMA, M.; HADID, A.; TAKESHIMA, H.; SHOTTON, J.; KOZAKAYA, T.; AND YAMAGUCHI, O., Facial deblur inference using subspace analysis for recognition of blurred faces. *Pattern Analysis and Machine Intelligence*, 33, 4 (2011), 838–845. (cited on page 25)

-
- OSHER, S.; BURGER, M.; GOLDFARB, D.; XU, J.; AND YIN, W., An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, (2005), 460–489. (cited on pages 92 and 108)
- PAN, J.; HU, Z.; SU, Z.; AND YANG, M.-H., Deblurring Face Images with Exemplars. In *European Conference on Computer Vision*, 47–62. (cited on pages xviii, xx, 8, 9, 26, 44, 57, 66, 68, 69, 75, 76, 77, 79, and 80)
- PAN, J.; HU, Z.; SU, Z.; AND YANG, M. H., Deblurring Text Images via L0 Regularized Intensity and Gradient Prior. In *Computer Vision and Pattern Recognition*, 2901–2908. (cited on pages xx, 21, 44, 77, and 79)
- PATI, Y. C.; REZAIIFAR, R.; AND KRISHNAPRASAD, P. S., Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, 40–44. (cited on page 36)
- PENG, Y.; GANESH, A.; WRIGHT, J.; XU, W.; AND MA, Y., RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence*, (2012), 2233–2246. (cited on pages xx, xxi, 94, 98, 100, and 107)
- PERONA, P. AND MALIK, J., Scale-space and edge detection using anisotropic diffusion. *Transactions on Pattern Analysis and Machine Intelligence*, 12, 7 (1990), 629–639. (cited on page 3)
- PIRINEN, O.; FOI, A.; AND GOTCHEV, A., Color high dynamic range imaging: The luminance–chrominance approach. *International Journal of Imaging Systems and Technology*, 17, 3 (2007), 152–162. (cited on page 104)
- PLÖTZ, T. AND ROTH, S., Benchmarking denoising algorithms with real photographs. *arXiv preprint arXiv:1707.01313*, (2017). (cited on pages xxii, xxiv, 113, 122, 123, and 129)
- PORTILLA, J.; STRELA, V.; WAINWRIGHT, M.; AND SIMONCELLI, E., Image denoising using scale mixtures of Gaussians in the wavelet domain. *Transactions on Image Processing*, (2003), 1338–1351. (cited on page 28)
- RAJAGOPALAN, A. AND CHELLAPPA, R., 2014. *Motion deblurring: algorithms and systems*. Cambridge University Press. (cited on pages 15 and 17)
- REED, S. E.; ZHANG, Y.; ZHANG, Y.; AND LEE, H., Deep visual analogy-making. In *Neural Information Processing Systems*, 1252–1260. (cited on page 131)

- RICHARDSON, W. H., Bayesian-based iterative method of image restoration. *JOSA*, 62, 1 (1972), 55–59. (cited on pages 16, 17, 18, and 20)
- RIZZI, A.; GATTA, C.; AND MARINI, D., A new algorithm for unsupervised global and local color correction. *Pattern Recognition Letters*, 24, 11 (2003), 1663–1677. (cited on page 2)
- ROTH, S. AND BLACK, M. J., Fields of experts. *International Journal of Computer Vision*, 82, 2 (2009), 205–229. (cited on pages xxi, xxii, xxiv, 39, 107, 108, 113, 114, 115, 118, 120, 121, and 122)
- SCHMIDT, U. AND ROTH, S., Shrinkage fields for effective image restoration. In *Computer Vision and Pattern Recognition*, 2774–2781. (cited on page 39)
- SCHULER, C. J.; HIRSCH, M.; HARMELING, S.; AND SCHÖLKOPF, B., Learning to deblur. *Pattern Analysis and Machine Intelligence*, 38, 7 (2016), 1439–1451. (cited on page 27)
- SHAN, Q.; JIA, J.; AND AGARWALA, A., High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, vol. 27, 73. (cited on pages 18, 20, 22, 24, 44, 49, 57, 68, 69, 70, 76, and 77)
- SHI, J. AND MALIK, J., Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22, 8 (2000), 888–905. (cited on page 3)
- SHI, J.; XU, L.; AND JIA, J., Discriminative blur detection features. In *Computer Vision and Pattern Recognition*, 2965–2972. (cited on pages xviii and 65)
- SIM, T.; BAKER, S.; AND BSAT, M., The CMU pose, illumination, and expression (PIE) database. *Automatic Face and Gesture Recognition*, (2002), 46–51. (cited on pages xviii, xix, 56, 62, 73, 76, and 94)
- SUN, J.; CAO, W.; XU, Z.; AND PONCE, J., Learning a convolutional neural network for non-uniform motion blur removal. In *Computer Vision and Pattern Recognition*, 769–777. (cited on page 9)
- SUN, L.; CHO, S.; WANG, J.; AND HAYS, J., Edge-based blur kernel estimation using patch priors. In *International Conference on Computational Photography*, 1–8. (cited on pages xix, 24, 25, 27, 57, 66, 68, 69, 70, 72, 75, 76, and 77)
- SUN, L.; CHO, S.; WANG, J.; AND HAYS, J., Good image priors for non-blind deconvolution. In *European Conference on Computer Vision*, 231–246. (cited on pages 9 and 35)

-
- SUN, L.; CHO, S.; WANG, J.; AND HAYS, J., Good Image Priors for Non-blind Deconvolution - Generic vs. Specific. In *European Conference on Computer Vision*, 231–246. (cited on page 19)
- TAI, Y.-W.; CHEN, X.; KIM, S.; KIM, S. J.; LI, F.; YANG, J.; YU, J.; MATSUSHITA, Y.; AND BROWN, M. S., Nonlinear Camera Response Functions and Image Deblurring: Theoretical Analysis and Practice. *Pattern Analysis and Machine Intelligence*, 35, 10 (2013), 2498–2512. (cited on page 44)
- TEODORO, A. M.; BIOUCAS-DIAS, J. M.; AND FIGUEIREDO, M. A. T., Image restoration with locally selected class-adapted models. In *Machine Learning for Signal Processing*. (cited on page 38)
- THOMAZ, C. E. AND GIRALDI, G. A., A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28, 6 (2010), 902–913. (cited on pages xix, xx, xxi, 74, 77, 97, and 102)
- TIKHONOV, A.; ARSENIN, V.; AND JOHN, F., Solutions of ill-posed problems. (1977). (cited on pages 3 and 16)
- TIMOFTE, R.; ROTHE, R.; AND VAN GOOL, L., Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1865–1873. (cited on page 114)
- TOMASI, C. AND MANDUCHI, R., Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, 839–846. IEEE. (cited on page 30)
- TORRALBA, A. AND OLIVA, A., Statistics of natural image categories. *Network: computation in neural systems (NCNS)*, 14, 3 (2003), 391–412. (cited on page 46)
- TROTT, T., The Effect of Motion of Resolution. *Photogrammetric Engineering*, 26 (1960), 819–827. (cited on page 43)
- VAN CITTERT, P., Zum einfluss der spaltbreite auf die intensitätsverteilung in spektrallinien. ii. *Zeitschrift für Physik*, 69, 5-6 (1931), 298–308. (cited on page 17)
- VAN GEMERT, J.; GEUSEBROEK, J.-M.; VEENMAN, C.; SNOEK, C.; AND SMEULDERS, A., Robust Scene Categorization by Learning Image Statistics in Context. In *Computer Vision and Pattern Recognition Workshop*, 105–105. (cited on page 46)
- VIGNESH, R.; OH, B. T.; AND KUO, C.-C. J., Fast non-local means (NLM) computation with probabilistic early termination. *Signal Processing Letters*, (2010), 277–280. (cited on page 97)

- VRHEL, M. J. AND TRUSSELL, H., Color correction using principal components. *Color Research & Application*, 17, 5 (1992), 328–338. (cited on page 2)
- WEISS, Y. AND FREEMAN, W. T., What makes a good model of natural images? In *Computer Vision and Pattern Recognition*, 1–8. (cited on page 108)
- WHYTE, O.; SIVIC, J.; AND ZISSERMAN, A., Deblurring Shaken and Partially Saturated Images. *International Journal on Computer Vision*, 110, 2 (2014), 185–201. (cited on pages 21 and 44)
- WIENER, N., 1949. *Extrapolation, interpolation, and smoothing of stationary time series*. (cited on pages 15 and 16)
- WOLBERG, G., 1990. *Digital image warping*, vol. 10662. IEEE computer society press Los Alamitos, CA. (cited on page 2)
- WOODS, J. AND INGLE, V., Kalman filtering in two dimensions: Further results. *Transactions on Acoustics, Speech, and Signal Processing*, 29, 2 (1981), 188–197. (cited on page 16)
- XU, J. AND OSHER, S., Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising. *Transactions on Image Processing*, (2007), 534–544. (cited on pages 92 and 108)
- XU, J.; ZHANG, L.; ZUO, W.; ZHANG, D.; AND FENG, X., Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising. In *International Conference on Computer Vision*, 1211–1218. (cited on pages 38 and 94)
- XU, L. AND JIA, J., Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, 157–170. (cited on pages 18, 20, 44, 66, 68, 69, 70, and 76)
- XU, L.; TAO, X.; AND JIA, J., Inverse Kernels for Fast Spatial Deconvolution. In *European Conference on Computer Vision*, 33–48. (cited on pages 21 and 44)
- XU, L.; ZHANG, L.; ZUO, W.; ZHANG, D.; AND FENG, X., Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising. (2015). (cited on pages 36 and 38)
- XU, L.; ZHENG, S.; AND JIA, J., Unnatural L_0 sparse representation for natural image deblurring. In *Computer Vision and Pattern Recognition*, 1107–1114. (cited on pages xix, 8, 44, 57, 68, 69, 70, 73, and 76)

-
- YAN, R.; SHAO, L.; CVETKOVIC, S. D.; AND KLIJN, J., Improved nonlocal means based on pre-classification and invariant block matching. *journal of display technology*, 8, 4 (2012), 212–218. (cited on page 31)
- YANG, J.; ZHANG, Y.; AND YIN, W., An efficient TVL1 algorithm for deblurring multi-channel images corrupted by impulsive noise. *SIAM Journal on Scientific Computing*, 31, 4 (2009), 2842–2865. (cited on page 18)
- YU, G. AND SAPIRO, G., DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, (2011). (cited on page 28)
- YUAN, L.; SUN, J.; QUAN, L.; AND SHUM, H.-Y., Image deblurring with blurred/noisy image pairs. In *ACM Transactions on Graphics (TOG)*, vol. 26, 1. (cited on page 49)
- YUAN, L.; SUN, J.; QUAN, L.; AND SHUM, H.-Y., Progressive inter-scale and intra-scale non-blind image deconvolution. In *Acm Transactions on Graphics (TOG)*, vol. 27, 74. (cited on page 18)
- YUE, H.; SUN, X.; YANG, J.; AND WU, F., CID: Combined Image Denoising in Spatial and Frequency Domains Using Web Images. In *Computer Vision and Pattern Recognition*, 2933–2940. (cited on pages 35, 36, 94, and 108)
- YUE, H.; SUN, X.; YANG, J.; AND WU, F., Image Denoising by Exploring External and Internal Correlations. *Transaction on Image Processing*, (2015), 1967–1982. (cited on pages 11 and 94)
- ZHANG, H.; YANG, J.; ZHANG, Y.; NASRABADI, N. M.; AND HUANG, T. S., Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *International Conference on Computer Vision*, 770–777. (cited on pages xix, 25, 77, and 78)
- ZHANG, K.; ZUO, W.; CHEN, Y.; MENG, D.; AND ZHANG, L., Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *Transactions on Image Processing*, (2017). (cited on pages xxii, 11, 39, 40, 108, 109, 111, 113, 116, 120, 121, 122, and 124)
- ZHANG, K.; ZUO, W.; GU, S.; AND ZHANG, L., Learning Deep CNN Denoiser Prior for Image Restoration. *Computer Vision and Pattern Recognition*, (2017). (cited on pages 39, 40, 108, 111, 113, 116, 120, 121, and 122)
- ZHANG, L.; DONG, W.; ZHANG, D.; AND SHI, G., Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, (2010), 1531–1549. (cited on pages 28, 31, 33, and 94)

ZHANG, W.; SUN, J.; AND TANG, X., Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, 802–816. (cited on pages xix, xxi, 45, 56, 72, 75, 94, and 101)

ZHONG, L.; CHO, S.; METAXAS, D.; PARIS, S.; AND WANG, J., Handling noise in single image deblurring using directional filters. In *Computer Vision and Pattern Recognition*, 612–619. (cited on pages 57, 68, 69, 70, 76, and 77)

ZORAN, D. AND WEISS, Y., From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, 479–486. (cited on pages 11, 19, 24, 25, 27, 35, 36, 37, 38, 57, 94, 112, and 113)